AD 745564

AFCRL-72-0092

TESTING FOR FAULTS IN CELLULAR LOGIC ARRAYS

By

DAVID A HUFFMAN

CONTRACT F19628-70-C-0168
PROJECT No. 4641
TASK No. 464104
WORK UNIT No. 46410401

Final Technical Report
Covering the Period 7 January 1970 to 6 January 1972

January 1972

Contract Monitor: Marvin E. Brooking
Data Sciences Laboratory

Approved for public release, distribution unlimited

Prepared for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS 01730

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Stanford Research Institute 333 Ravenswood Avenue Menlo Park, California 94025	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

TESTING FOR FAULTS IN CELLULAR LOGIC ARRAYS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Scientific. Final. 1 January 1970 to 6 January 1972 Approved: 1 March 1972

5. AUTHOR(S) (First name, middle initial, last name)

David A. Huffman

6. REPORT DATE	7a. TOTAL NO OF PAGES	7b. NO OF REFS
January 1972	104	3

8a. CONTRACT OR GRANT NO F19628-70-C-0168 b. PROJECT NO Project, Task, Work Unit Nos. 4641-04-01 c. DoD Element 62702F d. DoD Subelement 674641	9a. ORIGINATOR'S REPORT NUMBER(S) SRI Project 8487 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFCRL-72-0092

10. DISTRIBUTION STATEMENT

A - Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES TECH, OTHER	12. SPONSORING MILITARY ACTIVITY Air Force Cambridge Research Laboratories (LR) L. G. Hanscom Field Bedford, Massachusetts 01730

13. ABSTRACT

This report deals with the problem of determining the conditions under which it is possible to detect faults in two-dimensional cellular arrays with purely combinational logic and unilateral signal flow in both dimensions. The problem can be separated into two parts: the "excitation" problem and the "propagation" problem.

Novel procedures are developed for both of these problems. For the excitation problem ~~we have~~ determined what signals can be applied to cells arbitrarily deep within the array. For the propagation problem ~~we have~~ determined, separately for each of the two dimensions, how to propagate errors to the corresponding output boundary. Because the solution to each of these problems is expressed in terms of finite-state transformations, they can be combined. The resulting tables show whether or not faults of the various types can be tested for in cells arbitrarily remote from all the boundaries of a cellular array of arbitrarily large size.

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6801

UNCLASSIFIED
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Fault testing						
Fault diagnosis						
Cellular arrays						
Test generation						
Combinational logic networks						

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

AFCRL-72-0092

SRI Project 8487

TESTING FOR FAULTS IN CELLULAR LOGIC ARRAYS

By

DAVID A HUFFMAN

Stanford Research Institute
333 Ravenswood Avenue
Menlo Park, California 94025

CONTRACT F19628-70-C-0168
PROJECT No. 4641
TASK No. 464104
WORK UNIT No. 46410401

Final Technical Report
Covering the Period 7 January 1970 to 6 January 1972

January 1972

Contract Monitor: Marvin E. Brooking
Data Sciences Laboratory

Approved for public release, distribution unlimited

Prepared for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
BEDFORD, MASSACHUSETTS 01730

PREFACE

This final report covers the results of a small two-year research investigation, the objective of which was to attain a fundamental understanding of the inherent relationships between the internal logical structure and the input/output signal patterns that characterize the fault-free behavior of one and two-dimensional cellular arrays. This understanding is central to any solution to the problem of testing the very large-scale integrated semiconductor modules anticipated for highly reliable, future digital systems. Such modules are completely useless unless they can be thoroughly tested in advance, and reliability specifications require that such modules be readily testable in actual use.

The most complex class of cellular arrays treated in the study is the two-dimensional combinational array with unilateral signal propagation in both dimensional directions. This array is equivalent to a one-dimensional synchronous sequential array with unilateral signal flow between cells. Only single faults are assumed, although the fundamental nature of the investigation permits multiple faults to be handled as well, at least in principle. Emphasis has been placed throughout on questions of the inherent testability of prescribed one and two-dimensional arrays.

William H. Kautz
Project Leader

ABSTRACT

This report deals with the problem of determining the conditions under which it is possible to detect faults in two-dimensional cellular arrays with purely combinational logic and unilateral signal flow in both dimensions. The problem can be separated into two parts: the "excitation" problem and the "propagation" problem.

Novel procedures are developed for both of these problems. For the excitation problem we have determined what signals can be applied to cells arbitrarily deep within the array. For the propagation problem we have determined, separately for each of the two dimensions, how to propagate errors to the corresponding output boundary. Because the solution to each of these problems is expressed in terms of finite-state transformations, they can be combined. The resulting tables show whether or not faults of the various types can be tested for in cells arbitrarily remote from all the boundaries of a cellular array of arbitrarily large size.

CONTENTS

ILLUSTRATIONS

I INTRODUCTION

In this report we deal with problems associated with the detection of faults in arbitrarily large two-dimensional arrays of cells of combinational logic. We have assumed that all cells are constructed in the same way, that signals can pass from cell to cell in one direction only for each dimension, and that the array input and output signals can be applied to and observed only at the edges of the array. The purpose of this report is to develop new techniques applicable to arrays meeting the above conditions and not to derive exhaustive catalogs of possible cells and their fault propagation characteristics. With this in mind our attention has been focused on cells with binary inputs and outputs since the nomenclature problem is then simplified as far as possible; the techniques, nevertheless, can easily be seen to apply to cells with signals from larger alphabets. Furthermore, we have chosen to concentrate our attention in this report on treating in some depth a relatively few carefully chosen and representative examples of cell logic out of the multitude of examples considered during the period of this research. We shall do our best to point out the limitations of our techniques as well as their power, and to admit when we feel answers are still to come as well as when we feel problems have been completely solved.

An example of cell logic for a uniform cellular array with unilateral signal flow in each of its two dimensions is described in Figure 1.* The cell inputs are taken to be at the left ("west") and upper ("north")

---

*Figures are grouped at the end of this report.

edges of the cell and its outputs are at the right ("east") and lower ("south") edges. It is convenient to establish slightly different nomenclature for the signals possible at the west and east edges and the signals possible at the north and south edges. These are taken to be $\{0,1\}$ and $\{a,b\}$, respectively. It is also convenient to establish a shorthand notation for the four possible ordered pairs of cell input signals and a different notation for the four possible ordered pairs of cell output signals. These correspondences are:

$$A = (0,a) \qquad B = (0,b) \qquad C = (1,a) \qquad D = (1,b)$$

and

$$\alpha = (a,0) \qquad \beta = (b,0) \qquad \gamma = (a,1) \qquad \delta = (b,1) \quad .$$

(The last mode in the figure can arise only under error conditions, to be described subsequently.) A given cell logic is then described by a mapping from the cell inputs to the cell outputs. For our example this mapping is:

$$A \rightarrow \alpha \qquad B \rightarrow \alpha \qquad C \rightarrow \gamma \qquad D \rightarrow \beta \quad .$$

It is useful to have available matrices that show the dependence of the input or output signals associated with a cell upon the state of the input or output signals of the neighboring cells to the west and north. Four versions are possible and the matrices for our example are shown in Figure 2. Those in Figures 2(a) and 2(d) are especially useful. The first two matrices are derived by noting that the output signals associated with the neighboring cells (those outputs are shown in parentheses at the edges of the arrays) constitute the input signals to the cell under consideration, and then referring to the cell logic given in Figure 1. The last two matrices are derived from the first two by noting the mapping from the cell inputs to the cell outputs.

In Figures 3 and 4 are portrayed a 12 by 13 rectangular array comprised of cells with the logical properties of this example. (Ignore for a moment the symbols in the lower-right of some of the cells.) We note that the signals at the input boundary of the array determine uniquely the states of the interior cells of the array as well as the signals at its output boundary. The information displayed in Figures 2(a) and 2(b) was used in deriving these two representations. Assume now that a fault occurs in the cell fourth from the north and third from the west and that this fault changes the signal at the east edge of that cell from 1 to 0. That is, the output of that cell changes from γ to α. This change will cause changes in the outputs from other cells in the array (see especially Figure 4). In this case the fault causes changes that finally appear at the east boundary of the array. We shall say that the fault has caused errors that propagate to that boundary.

We find that faults and the associated error propagation are most easily seen in terms of the cell output signals (see for example, Figure 4). The question of how the cell states depend upon the array inputs, on the other hand, is perhaps more naturally associated with the cell input signals (for example, see Figure 3). These latter signals are usually the ones used by others (notably Kautz[1] and Menon and Friedman[2]) in their discussions of the "tesselation" problem; that is, of the problem of determining what patterns (usually with some periodic or regular structure) of cell states in the array are compatible with possible array input signals. We shall not restrict ourselves in this way and shall refer, generally, to those issues as the "excitation" problem.

---

*References are listed at the end of this report.

In our approach to the study of excitation problems we shall assume, in effect, that the faulty cell (usually only one) is arbitrarily far from the input edges of the array. Similarly, in studying the error propagation problem we shall assume that the faulty cell is arbitrarily far from the output edges of the array. If a cell can be excited in a certain way when it is far from the input edges of an array then it can certainly be excited in that same way when one or more columns of cells are removed from the west boundary of the array or when one or more rows of cells are removed from the north boundary of the array. Similarly, if error propagation paths exist from a faulty cell arbitrarily remote from the output edges of an array, those paths certainly still exist when one or more rows of cells are removed from the south boundary of the array or when one or more columns are removed from the east boundary of the array. Because we wish to consider the excitation and propagation problems for cells arbitrarily placed in arbitrarily large arrays we must somehow consider only excitation patterns that can exist arbitrarily remote from the input boundaries of the array and error propagation paths that are guaranteed to reach array output boundaries arbitrarily far from such cells.

For nearly all of this report we will assume that the input signals to the array actually occur on a hypothetical diagonal boundary that is somewhere to the northwest of (perhaps tangent to) the input boundaries of the array (see Figure 5) and that the intercell signals must propagate to a hypothetical diagonal that is somewhere to the southeast of (perhaps tangent to) the output boundaries of the array. The rectangular array (or any other array which has a convex boundary) is then contained between those newly established input and output diagonals. Certainly any excitation signals that can be applied to any cell contained between these diagonals by freely choosing the signals applied to the input diagonal can also be applied to that same cell if we are allowed to choose the

inputs to the array input boundaries. Similarly, any error propagation paths that reach the output diagonal would certainly have reached the array output boundaries.

The cell input signals on that diagonal that contains the possibly-faulty cell are conveniently thought of as sequences of input signals that proceed from southwest to northeast. As we have indicated, we shall be especially interested in the set of sequences that can exist on diagonals which are "deep" in the array (that is, arbitrarily remote from the input diagonal). That set of sequences will be termed the "deep-diagonal set" or, more briefly, the "deep-set." The next several sections of this report will consider how to determine these sets and what properties they have.

In deriving the deep-diagonal sets it was necessary to consider in some detail the transformation between the sequence of signals on one diagonal and the sequence of signals on an adjacent diagonal. This transformation, as we shall see, is a finite-state transformation. Several other transformations were considered and studied in some detail during the course of this research but all were rejected in favor of the diagonal-to-diagonal transformation. For instance, the input to a rectangular array can be considered to be a semi-infinite sequence of composite symbols that starts at the northwest corner of the array and has as the component parts of its $\underline{k}$th member the $\underline{k}$th symbol from that corner on the west boundary and the $\underline{k}$th symbol from that corner on the north boundary. A finite state transformation converts such a sequence to another such sequence one cell layer into the array from the input boundary.

It is possible that a deep set may not contain some particular cell input symbol (A, B, C, or D). In that case we would not have to consider the propagation of errors from a fault in such a cell. In

this our approach differs somewhat from that of Kautz. He termed "fully testable" only those cells that, when placed in an arbitrary location in an arbitrarily large array, could have applied to them any one of the cell inputs (and that, for any one of the possible single-cell faults, could propagate the corresponding errors to an arbitrarily remote output boundary).

In our approach we admit that in finite sized arrays it is possible both for a cell to lie close enough to the input boundary that it can be excited in a way which would not be possible if the cell were farther away from that boundary, and for that cell to lie close enough to the output boundary that errors could propagate to it when that would not be possible for a more distant boundary. Only if both conditions exist would a faulty cell be more troublesome in use in an array producing erroneous outputs than our method would indicate. An extreme example would be to consider an array consisting of a single cell that could be excited in one of the four possible ways and that, when faulty, would always (via a zero length path!) propagate its error to the output boundary. Even in arbitrarily large rectangular arrays there are cells at and near the southwest and northeast corners of the array that can be excited, and from which errors can be propagated in special ways that are not possible for cells near the interior of the array.

In general, it is possible that excitation of and error propagation from cells in "small" arrays (or at or near these corners in "large" arrays) may not be treated by our approach. The only alternative, it seems to us, would be to consider both the excitation and propagation problems for a given cell as functions of distance of that cell from all four array boundaries. We would also, in effect, have to consider separately all possible sizes of arrays. In order to obtain general results for arbitrary sized arrays we therefore believe it is quite reasonable to limit our study in the way that we have.

The emphasis in the research reported here is upon the detection of faults rather than upon the location of faults. That is, we shall concentrate our attention upon procedures that allow us to guarantee that a cell, if faulty, will cause a change of output boundary signal rather than upon the problem of determining which cell is faulty, although the possibility of answering the latter question is implicit in the former issue.

II DERIVATION OF THE DEEP-DIAGONAL SET USING THE CELL INPUT SYMBOLS

In this section we shall derive, for the example used in the introductory section, the deep-diagonal set of sequences. We shall assume that all possible sequences of the cell input symbols (A, B, C, and D) are possible on the input (or "0-level") diagonal. By viewing the mapping of this universal set of sequences onto the set of sequences on the next (or "1-level") diagonal (that is, the next diagonal to the southeast) as a finite-state transformation we can express the constraints among the symbols on that next diagonal in state table (or state-diagram) form. By thinking of this new set of sequences as the input to the same diagonal-to-diagonal transducer we derive, again in tabular form, the constraints on the new diagonal. In general, we can derive the set of sequences that is possible at the (k + 1) st-level diagonal by applying the set of sequences at the kth level to the transducer (see, for instance the example in Figure 6(b).

At each stage of this process the set of sequences under consideration is a regular set; that is, one that can be described by a finite state table or equivalent state diagram. (For a full discussion of the correspondence between regular sets and finite state machines see, for example, Hennie.[3]) In this first example the set that is the limit as $k \to \infty$ will be obvious. This limiting set is the deep-diagonal set we are after.

Our point of departure is the matrix first given in Figure 2(a) and repeated in Figure 6(a). From it we have derived and displayed an example of a transformation from one diagonal to the next [Figure 6(b)]. In this figure are included the intercell signals (0 or 1) that we shall

Preceding page blank

think of as the "states" in the diagonal-to-diagonal transducer. The sequence of symbols on the first of these diagonals will be thought of as the input sequence to the transducer and the sequence of symbols on the second diagonal as the corresponding output sequence.

In Figure 6(c) we have shown a representative step in the transformation. The initial state is the 1 symbol, the symbol A (on the first diagonal) is the input symbol, the symbol C is the corresponding output symbol, and the next state is given by the 0 symbol. In general, both the next state and the output symbol are functions of the initial state and input symbol.

From the matrix we can see that if the input is A, B, or D the next state is 0 and if the input is C the next state is 1. This information and that about the dependence of the output upon the initial state and input are shown in the table of Figure 6(d). The entry in the lower left corner of this table corresponds to the example cited above. This table shows the diagonal-to-diagonal finite state transducer (call it "T") that we have referred to earlier.

If the universal set of sequences is applied to a transducer T, as is possible at the 0-level, the sequence at the 1-level will, in general, be constrained. In order to see what restrictions apply in our example we show in a new table [Figure 6(e)] how the next state depends upon the initial state and the output symbol. (We shall call this table the "output table.") For instance, if the initial state is 1 the output can be D and the next state must then be 0. If the initial state is 0 the output cannot be C, and therefore there is no corresponding entry. If the initial state is 0 and the output is A the next state could be either 0 or 1. The first two rows of the new table have been derived using the reasoning illustrated above.

If we have analyzed a possible output sequence and found that the state is either 0 or 1 and, for instance, that the next symbol is C, then the ambiguity about the state still remains. If, on the other hand the symbol had been B the next state would have to be 0. The row containing these data has been labelled 01 and its entries can be found by taking the union of the entries (in the appropriate column of the table) from the first two rows. We can further observe that at any given point in a sequence of output symbols the state is either known to be 0 or is ambiguous (0 or 1). For this reason these two rows are called the "accessible" rows and have been checked in the output table.

In order to avoid unnecessarily complicated nomenclature in future steps of our procedure it is convenient to relabel 0 and 01 as 1 and 2, respectively, in the output table. The resulting table and equivalent state-diagram are shown in Figure 6(f) and 6(g). The latter diagram tells us that at the 1-level diagonal it is not possible to follow either a B or D symbol by C or D; B or D can be followed only by A or B. The two states of the diagram show just enough about what the past history of a diagonal sequence has been to determine what constraints apply next. From now on it will be unnecessary to try to recall how these states were associated with the states of the transducer T.

We determine the 2-level diagonal set by applying the 1-level set as input to T. A convenient clerical procedure involves the use of a new table [see Figure 7(a)]. The states of that table are labelled with two digits. The left digit shows the state of the 1-level set that is applied to T. The right digit is the state of T itself. From the state of the 1-level set and the diagonal symbols (which constitute the input to the 2-level diagonal) we determine which transitions are possible and, for those, what the new state of the 1-level set is. This information is contained in the table of Figure 6(f) and is repeated as the left digits of the table entries. From the possible initial states of T

and the diagonal symbols we determine the new states of T and the output symbol. These data are available in Figure 6(d) and are shown as the right digits of the entries and the superscripts,

Before proceeding further it is convenient to note which states can actually be reached (observe that no transitions lead to the state with the 11 label) and to relabel these states. The resulting table [see Figure 7(b)] shows how the symbols at the 1-level (the input symbols) are transformed to the symbols at the 2-level (the output symbols).

By the same procedure used earlier we now determine the output table [Figure 7(c)] associated with this transformation. For instance, if the state is 2 and output symbol is A the next state is 1, 2, or 3. This ambiguity is shown in the output table by the 123 entry in row 2, column A. In order to determine how the ambiguity among states is dependent upon the next output symbols we again form the union of entries from the rows associated with the possible states. For example, in the A column of the row labeled by 12 we place the label 123 since the labels in that same column from the rows labeled 1 and 2 are 12 and 123, respectively. It can be seen that the rows labeled 2 and 3 are inaccessible. The accessible rows have been checked and given new labels in the table [Figure 7(d)] and corresponding state diagram [Figure 7(e)]. These show the constraints for the 2-level diagonal set.

A repetition of the procedure described above leads to the 3-level set shown in Figure 8(a). In the example under consideration it is comparatively easy (by repeated application of the procedures which have been described) to see that the k-level set is that given in Figure 8(b). We can conclude from this diagram that, for example, two D symbols on a diagonal which is k levels to the southeast of the input diagonal must be separated by at least k A symbols, and that two B symbols must be separated by at least k-1 A symbols.

In the limit (as $k \to \infty$) a B or D symbol can be followed only by A symbols. The state diagram representative of this statement is that given in Figure 8(c). This diagram describes the deep-diagonal set referred to earlier. It can be verified (using the same techniques which we have illustrated in Figures 6 and 7) that when this limiting set is applied to the transducer T the resulting output set is that same deep-diagonal set. The steps in this verification are shown in Figure 9. The relabeling of the two accessible states in the output table yield the deep-set [the same one shown in Figure 8(c)].

It is interesting to examine the pattern of cell states given in Figure 3 with the state diagram of Figure 8(c) in mind. The basic pattern shown is one that could be extended to an indefinitely large array. Its main features are a single B with all D's directly above it, a uniform subpattern of C's in the quadrant to the northwest of the B, a uniform subpattern of A's in the quadrant to the northeast of the B, and rows of A's or C's in the half-array below the B. The reader should proceed to the northeast along each diagonal in that pattern and determine for himself a corresponding path through the state diagram of Figure 8(c).

It should be clear that each path through the state diagram represents a sequence of diagonal (input) symbols that could be sustained arbitrarily far into the array from the input diagonal. The reader might wish to determine how to produce that sequence by providing some appropriate excitation to the input diagonal. Unfortunately it can be shown that, for this purpose, the description of the transformation between the sequence on the input diagonal and the sequence on another diagonal further into the array requires, in general, a number of states exponentially related to their separation.

It should also be clear that each deep-set sequence on a given diagonal can be produced by at least one other deep-set sequence on

on the preceeding diagonal. As a consequence of this fact any deep-set sequence which can be produced far inside the array can be produced by exciting the input diagonal by another (at least one) deep-set sequence. Therefore, in order to test cells deep inside the array, only deep-set sequences need ever be applied to the input diagonal.

A possible procedure for determining the deep-set sequence (or sequences) that can cause a given deep-set sequence on the next (southeast) diagonal is straightforward but somewhat complicated. The transformation in Figure 9(b) is instrumental in that search. We assume that the sequence on that latter diagonal is

--- C, C, A, C, A, A, A ---

We construct a diagram showing the three possible states in the transformation mentioned above, once for each step of the sequence (see Figure 10). From the transformation and a knowledge of each of the desired output symbols on the diagonal we display the transition that could cause that output symbol and label the transition with the input symbol associated with it. Those paths through the resulting diagram represent the possible options available for the input sequence. In our example the input sequences are

--- C, A, C, (followed by a sequence of A's containing, at most, one B).

Any proposed procedure for determining what sequence or sequences to apply to one diagonal in order to cause a specified sequence on the next diagonal must, in effect, be equivalent to the search illustrated above. It should be apparent that the search for input sequences can be a quite complex one even when the specified output sequence is on the next diagonal. If the diagonals were separated more widely it would seem that the necessary procedure would be so complicated as to be quite unwieldy.

For these reasons we shall content ourselves with determining the deep-diagonal set, knowing that any excitation that can be produced for cells deep inside the array can be produced by restricting the excitation on the input diagonal to sequences from that same deep-set.

III DERIVATION OF THE DEEP-SET USING CELL OUTPUT SYMBOLS

The deep-diagonal sets can also be derived in terms of the cell output symbols (α, β, γ, and δ) using methods analogous to those of the preceding section. For a starting point we choose the matrix of Figure 2(d). From it we obtain the table expressing the transformation from one diagonal sequence of cell output symbols to the sequence of those same symbols on the next diagonal [see Figure 11(a)]. The transducer that is appropriate for this task we shall call τ, to distinguish it from the one of the preceding section.

By listing the dependency of the next state (or states) upon the present state and the output symbol we get the output table shown in Figure 11(b). There is only one accessible state in this table. The corresponding row tells us that at the 1-level diagonal the symbol δ cannot occur; there are no further constraints on that diagonal.

When the set of constrained input sequences at the 1-level is applied to τ, the table given in Figure 11(c) is the result. The output table [Figure 11(d)] and a relabeled version of it [Figure 11(e)] follow directly. The 2-level set can, in turn, itself be applied to τ and the 3-level set can be derived. (This derivation is not shown.) By repeating the same procedure it soon becomes evident that the k-level set is that described in Figure 11(f).

As $k \to \infty$ the length of the chain of transitions associated with the α symbol increases proportionately. In the limit it is obvious that a β symbol can be followed only by an all-α sequence. The situation in the limit is described by the state diagram of Figure 11(g). Consequently that diagram describes the deep-set. It can be verified that that set,

Preceding page blank

when applied to the transducer τ as input yields as the output that same deep-set. The critical steps in that verification procedure are given in Figure 12 and they parallel the steps given in Figure 9, described earlier. (At the last step ([Figure 12(e)] the bottom two rows contain identical entries and thus describe indistinguishable states. They have therefore been replaced by a single state.)

From the new description of the deep-diagonal set we can see that no δ symbols ever occur, that at most one β can occur on a diagonal, and that if it does it can be followed only by α's. The reader should examine the various diagonal sequences shown in Figure 4 to prove to himself that they are each from the deep-set.

It is possible [by using the table of Figure 12(c)] to display a search procedure analogous to the one shown earlier in Figure 10 that demonstrates how to determine the sequence or sequences on one diagonal that can lead on the next diagonal to a sequence from the deep-set. This will be left as an exercise for the reader. As before, however, we are already assured that such sequences from the deep-set always exist.

It is clear that the two representations of the deep-diagonal set [Figures 8(c) and 11(g)] must somehow be equivalent. The remainder of the section will be devoted to demonstrating how to convert from one form to the other.

First we shall take the description of the deep-set in terms of the cell output symbols and change it to the description in terms of cell input symbols. Information important in performing this conversion is the deep-set of Figure 11(g) and the matrix of Figure 2(b). Recall that that latter matrix showed how the input symbol of a cell depended upon the output symbols of the cells to its west and to its north. That

is, we can determine from a knowledge of a pair of two consecutive cell output symbols on one diagonal an input symbol on the next diagonal.

The top portions of Figure 13(a) and (b) probably give us the best visualization of this transformation applied to the deep-set. We replace the four transitions of the state diagram for the deep set by nodes in a new diagram and give these nodes labels which tell us both what node the transition came from and what cell output symbol was associated with the transition. This cell output symbol is to be the first of the pair of consecutive symbols referred to in the paragraph above.

The cell output symbols (α, β, γ, or δ) associated with each transition in the new diagram correspond to the second of the pair of consecutive symbols. There is a transition from a node "i" to node "j" of the second diagram if and only if in the first diagram the transition "i" could be followed by transition "j." Therefore the pair of consecutive symbols in the first diagram corresponds in the second diagram to the symbol in the node label and the symbol on a transition which comes from that state. This pair of consecutive symbols determines uniquely [see Figure 2(b)] the appropriate cell input symbol (A, B, C, or D) to be associated with the transition.

As an example, we note in the first diagram that the β-labeled transition from state 2 (which enters state 1) can be followed only by a α-labeled transition. Thus in the second diagram a corresponding transition leaves the node 2β and it can only be labeled α. This transition is directed to the node 1α since (see the first diagram) the β-transition from state 2 yields state 1 and since the second of the pair of consecutive symbols was α. Because the pair of consecutive symbols was β, α the proper cell-input symbol for the transition is A.

The diagrams in equivalent tabular forms also appear in the figure. We note [Figure 13(b)] that the two components of the row label determine

uniquely the first components of the entries within the table and that the identity of these components can be found directly from the deep-set table. The second components of the entries are identical to the symbols heading the column. The superscript for an entry in a given row and column is determined from the second component of the row-label and the column-label, and the matrix in Figure 2(b).

The new state diagram and the cell input symbols on its transitions contain the information about the deep set we are after. As a preliminary step we note that states 1α and 2β are indistinguishable. We then, for convenience, relabel the states of that diagram [Figure 13(c)]. Finally we derive the corresponding output set. Only two states in it are accessible; these have been indicated by the usual check marks. These two rows can easily be seen to be equivalent to those of Figure 8(c), the result we were seeking.

By an analogous procedure we can convert a description of the deep-set using cell input symbols into the one using cell output symbols (see Figure 14). In this case the matrix that is useful in showing us how a cell output symbol on one diagonal depends upon the pair of consecutive input symbols on the preceding diagonal is the one shown in Figure 2(c).

An additional technique is available for the generation of the deep-set in terms of the cell output symbols. We first obtain the deep-set sequence in terms of the cell input symbols. Then, we replace these symbols on the transitions by the output symbols they map onto. (In our example $A \rightarrow \alpha$, $B \rightarrow \alpha$, $C \rightarrow \gamma$, and $D \rightarrow \beta$.) In general this new state diagram (or equivalent state table) will be nondeterministic. That is, two or more transitions associated with the same symbol may leave some of the nodes. By the use of standard techniques (see Hennie[3]) this state diagram may be replaced by an equivalent deterministic state diagram.

In the next section we will break away from the single example of cellular logic we have exploited in such detail up to this point. There we will attempt to give some perspective as to how difficult the determination of deep-sets can be.

IV EXAMPLES OF DEEP-DIAGONAL SETS OF SEQUENCES

In this section are summarized for comparative purposes the deep-sets which have been derived for a representative sampling of cell logic specifications. They range from the easiest and most obvious to the most difficult that have been studied. The basic methods which were used in their derivation are those that have been elucidated in such detail for our example of the earlier sections. Each of the new examples will be referred to by a four-symbol code that is the ordered set of the four cell outputs that result, respectively, from the four possible cell inputs (A, B, C, and D, in that order).

The example of Figure 15 (encoded $\beta\ \delta\ \gamma\ \alpha$) is one in which all four possible output symbols can occur. It follows that for each cell output symbol there exists a corresponding cell input that will cause that cell output. It is obvious that for every conceivable sequence of cell outputs on an array diagonal there is a (unique) corresponding sequence of cell inputs. Since these inputs are the outputs from the cells on the preceding diagonal we can conclude that every sequence of cell states can exist on diagonals arbitrarily deep in the array, and that once a sequence of cell states on one diagonal has been specified the sequence of states on all other diagonals (both preceding and following the given diagonal) is uniquely determined. Consequently, a change of cell output because of a fault anywhere in the array must necessarily propagate to all successive diagonals.

In the next example (Figure 16) only two cell outputs (β and γ) are possible but on a given diagonal these outputs can occur in any sequence. By recalling Figure 2(a) (which is valid for all examples of cell logic) we note that for every possible cell input (A, B, C, or D) there exists

Preceding page blank

a consecutive pair of cell outputs on the preceding diagonal that will give the specified cell input. Specifically $\beta\ \gamma \rightarrow A$, $\beta\ \beta \rightarrow B$, $\gamma\ \gamma \rightarrow C$, and $\gamma\ \beta \rightarrow D$. The corresponding deep-set is the second one shown in the present figure. Consequently we conclude that any cell of the array can be put into any one of the four possible input states by at least one choice of signals on the input diagonal from among the sequences in the deep-set.

From the diagram showing the deep-set we can see, for instance, that any given diagonal can have the periodic sequence of cell inputs - - - A C D B A C D B A - - -. It is <u>not</u> true that this periodic sequence can exist on <u>all</u> diagonals simultaneously. The reader will find it instructive to discover for himself what diagonal sequences are possible both preceding and following the periodic sequence specified above.

We can also observe in the $\gamma\ \beta\ \beta\ \gamma$ example that a change of the west component of the cell input always causes a change of the east component of the cell output. Similarly, a change of the north component of the input always causes a change in the south component of the output. We conclude, therefore, that any error signal from a fault involving the east output of a cell will propagate directly to the east in the array, and that any error signal from a fault involving the south output of a cell will propagate directly to the south in the array. Thus every conceivable type of fault occurring in a single cell will reveal itself by some change in signal on the output diagonal.

In the $\alpha\ \gamma\ \gamma\ \alpha$ example (Figure 17) the associated deep-sets are shown. We note that only two of the possible cell input states can occur deep in the array; cells in the interior of the array cannot be put into states B or D. A simple analysis of this example also reveals that an error on the east edge of a faulty cell propagates directly to the east and that (independently) an error on the south edge of a faulty cell causes propagation of errors directly to the east in the row of

cells directly below the faulty cell. Thus again all possible faults can be detected at the output diagonal even though it is not possible to excite cells deep in the array with either input B or input D.

From here on in this section we shall not attempt to comment on the error propagation problem since that topic is reserved for the later sections and is usually much more difficult than it was for previous examples.

In the $\gamma\ \beta\ \beta\ \gamma$ and $\alpha\ \gamma\ \gamma\ \alpha$ examples above and in the $\alpha\ \gamma\ \beta\ \gamma$ example of Figure 18 it is easy to demonstrate that the deep-set is the same as the 1-level set. In general we may have to compute a longer sequence of sets before we are certain what the deep-set is. (The example of the first section showed that it is possible for the deep-set to exist only as a limit of the sequence of k-level sets as $k \to \infty$.)

Cells of the type shown in Figure 19 compute the NOR-function of the cell inputs and deliver the result of this computation to both output edges of the cell. This ($\delta\ \alpha\ \alpha\ \alpha$) example is one for which the deep-set is the same as the 2-level set. It can be seen that any cell of an array can be put into any one of the four possible cell input states.

The $\gamma\ \alpha\ \beta\ \beta$ example of Figure 20 required a still more lengthy procedure before it was certain what the deep-set was. For that example the deep-set is the same as the 3-level set. Again it can be observed from the state diagram that any cell of an array can be put into any state.

When a particular cell structure leads to an infinite sequence of distinct k-level sets the problem of discovering the set that is the limit may or may not be an easy one. In the first such example [see Figures 8(b) and 11(f)] the limit was relatively easy to deduce. In our next example the limit is not so obvious.

In Figure 21 are displayed the results of calculating a sequence of k-level sets for the $\alpha\ \beta\ \beta\ \gamma$ example. This example is a "half-adder:" that is, the signal on the south edge of the cell is the "exclusive-OR" of the two cell inputs, and the signal on the east edge is the "AND" of these inputs. The limit (deep) set shown may or may not be apparent to the reader at first glance A technique which is often helpful (as it was in this case) is to consider the description of the set of "reversed" sequences. (It is well known that if all sequences in a regular set of sequences are reversed the result is another regular set.)

The table of Figure 22(a) corresponds to the 5-level set shown in state diagram form in Figure 21. The set which is the reverse of this set may be thought of as associated with that state diagram after all of its transition arrows have been reversed. The first five rows of the table in Figure 22(b) are a representation of that reversed diagram. For instance, in the original table we see that there are four β-labeled transitions which lead from states 2, 3, 4, and 5 to state 2. In the modified table we have indicated (in the β-column) that there are to be transitions from state 2 to states 2, 3, 4, and 5. The remaining entries in the first five rows of the table have been completed using the same reasoning.

We observe that it is possible to have an ambiguity about the state if the last symbol was a α and the last state was 5. This ambiguity is expressed by the two-component designator, 45. If the next symbol is α the new ambiguity (345) among the states may be found by taking the union of the state designators in column α and rows 4 and 5. Each possible new ambiguous possiblity has been documented in the lower half of the table. The five checked rows show which are accessible. The state diagram of Figure 22(c) corresponds to these five rows and the entries contained therein.

It is fairly easy to show that (for $k \geq 3$) the reversed k-level sets are described by diagrams like that shown here with the left-most chain of α's having a length of k-2. The limit for this sequence of reversed sets is then obviously the one given in Figure 22(b). We note that any sequence of symbols that is possible starting with the left node of this diagram is also possible starting with the center node. Thus we may eliminate the left node and the arrows that leave it without affecting the set of sequences we wish to describe. The remaining part of the state diagram, with its arrows re-reversed, is the one which we displayed in Figure 21.

The state-diagram visualization of both the k-level sets and their reverses is no guarantee that the limit will be easy to find. In Figure 23 we have shown an example that was particularly troublesome. The limit became fairly clear only after the sequence of sets up to and including the case k = 7 were displayed. The reversed sets were equally difficult to comprehend. The set that was thought to be the limit was then verified using the techniques illustrated in Figure 12.

V A STRAIGHTFORWARD APPROACH TO THE ERROR-PROPAGATION PROBLEM

Having found a mechanism for describing all of the diagonal sequences that can exist deep in the interior of a two-dimensional logical array, our next task is to determine under what conditions the errors from faulty cells on one diagonal can propagate to the output diagonal. This problem has been found to be inherently quite difficult, since the errors due to a fault can propagate through the array in very complex ways. For instance, errors that have propagated along two different paths and have arrived at both the west and north edges of a cell within the array may cancel each other at that cell. Because of this possibility one must somehow keep track of <u>all</u> of the paths along which errors can propagate from the given fault in order to determine whether or not at least one of these paths will ever reach the output boundary.

One possibility, of course, is to use the deep-sets and the inter-diagonal transformations we have derived to determine possible two-dimensional patterns of cell-input and corresponding cell-output symbols. We could then recalculate these patterns for whatever fault-type was of interest. It would be immediately apparent, once the new pattern had been computed, whether or not the signals at the array output boundary had changed and therefore whether or not the error had propagated to that boundary. This approach, though valid, would be extremely inefficient because it would require, for each excitation pattern, a recalculation of that pattern for each faulty-cell location and type of fault.

In this section we shall show what may be considered to be an absolutely straightforward and obvious approach to the propagation problem, although it took a significant amount of time and the study of many alternate approaches to decide that this was the case. It is based

 Preceding page blank

on the fact that if an error initiated at an arbitrarily placed cell in an array is to propagate to the boundary, then it must affect the sequence of signals at at least one cell along each diagonal further into the array. On the other hand the cessation of error propagation is associated with the possibility that two (or more) different sequences can lead to a common sequence on the next (or some other succeeding) diagonal. Consequently it is useful to approach the associated transformation problem directly. That is, we shall examine the transformation from the sequences on one diagonal to sequences on a subsequent diagonal as a function of the degree of separation of these diagonals.

As a working example of cell logic we return to one ($\gamma\ \alpha\ \beta\ \beta$) that was used in the last section of this report (see Figure 20). In Figure 24(a) we give the diagonal-to-diagonal transducer, τ. By, in effect, entering this table <u>twice</u> in succession we can determine the transformation from the output symbol of a cell on one diagonal to the output symbol of the cell that is directly south of it on the second diagonal away. An illustrative example is given in Figure 24(b). From τ we determine that, when the initial state is 0 and the input symbol is 1, then the next state is 1 and the output symbol is α. This output symbol becomes the input symbol for the second use of the table. If we assume that the initial state for this second use is also 0, then the corresponding next state is 0 and the output symbol is γ. The net effect is that when the (composite) input state is 00 and the input symbol is δ the next (composite) state is 10 and the output symbol is γ.

A table representing these transformations is given in Figure 24(c). The entries in that table (referred to as τ^2, the "<u>product</u>" of the earlier table and itself) are the next-state components and (shown as superscripts) the pair of output symbols mentioned above The second of these output symbols is the one of interest; the first is a convenience in deriving the table. The example shown in Figure 24(b) corresponds

to the entry in the upper right corner of the new table. In deriving the τ^2 table we compute first the left components of the entries directly from the left components of the composite state and the input symbol. We then compute the right components of the entries from the output (now input) symbol just computed and the right component of the composite state. (It would be possible to derive in one step, using an obvious extension of this procedure, tables representing τ^3, τ^4, τ^5, etc. It is apparent that the table representing τ^j would have 2^j rows.)

This same table, renumbered, is shown in Figure 24(d). We note that, in this case (it does not happen in general), the output symbols associated with a transition depend only upon the initial state. The corresponding state diagram has all arrows leaving a given node labelled with the same output symbol. It is useful to modify this table (or the corresponding state diagram) by labelling the transitions entering a given state (rather those leaving that state) with the symbol associated with the state. This modification corresponds to shifting the output sequence on its diagonal by one unit to the southeast. (The sequence is otherwise unchanged.) The new table, with its outputs shifted in this manner, is given in Figure 24(e). In this new table we see that states 1 and 2 are equivalent, as are states 3 and 4. The appropriately simplified and relabelled table is given in Figure 24(f).

We can now find the τ^3 transformation by taking the product of the τ^2 table we have derived and the original τ table. [See Figure 25(a).] (The product may be taken in either order since it is clear that that operation is associative.) The initial states have two components, one from among the states of τ^2 and the other from among those of τ. Again, we begin by computing the left components of the entries (using the τ^2 table). From those output (now input) symbols and the right components of the initial states we compute the right components of the entries (using the τ table). In this case only three states are accessible.

The relabelled table is shown in Figure 25(b). (No shifting of the output sequence was possible.)

Reapplication of this procedure (using the τ^3 and τ tables) results, after a shifting of the output sequences, in the τ^4 table of Figure 25(c). Still another application of the procedure yields (after a shifting) the τ^5 table. When the τ^6 table is computed it can be seen to be the same as the one for τ^3. It is then apparent that, for this example, $\tau^{j+3} = \tau^j$ $(j \geq 3)$.

The tables for τ^3, τ^4, and τ^5 are obviously not isomorphic to each other. However, it is readily determined that they have a common <u>output</u> set. This output set is the deep-set derived earlier (see Figure 20). In fact we could have determined the deep-set from the sequence of tables describing τ, τ^2, τ^3 - - -. However if it is only the deep-set itself that is of interest then determining this sequence of transformations would be a needlessly cumbersome way of arriving at that more limited goal.

The reader should <u>not</u> conclude that the number of states required for a description of τ^j will, <u>in general</u>, reach a finite limit as $j \to \infty$. In fact, in only a very few of the cell structures studied was it possible to achieve such a limit. Therefore the present example represents an exceptional case rather than the rule. It is nevertheless instructive because it furnishes a first example of a state table that we can interpret to obtain information about error propagation.

We shall first concentrate our attention on the transformation described by Figure 25(b). It describes the transformation from an arbitrary sequence of cell output symbols on one diagonal to the sequence it causes on diagonals separated from it by 3, 6, 9, - - - units. Assume that the sequence of symbols on the first diagonal corresponds to state 3, that the symbol at the next cell is nominally α, and that a fault

occurs in that cell which changes the cell symbol to β. (We shall call this an "α-β fault.") We can see from the table that in this instance the fault cannot be detected since both α and β lead to γ on the second diagonal and to the same next state (2). We note also that the reasoning for a β-α fault would be the same as for the α-β fault.

If the initial state had been 1 and the fault of type γ-δ it would have been detected since γ or δ on the first diagonal leads to α or γ on the second diagonal, respectively, even though each leads to the same next state.

If the initial state were 2 and the fault of type α-β the symbol produced on the second diagonal would be β regardless of whether the fault were present or not. However the next state can be seen to be 1 without the fault and 2 with the fault present. Because states 1 and 2 are not equivalent there must be some continuation of the sequence on the first diagonal that will let us distinguish between them. That is, there must be some continuation of the first sequence that will ultimately lead to differing symbols on the second diagonal. By examining rows 1 and 2 of the table we see that this distinction can be made at the very next step because, regardless of the next symbol on the first diagonal, the corresponding symbols on the second diagonal differ from each other. Thus errors from faults of type α-β will always propagate to the second diagonal when the initial state is 1.

If the initial state had been 2 and the fault of type α-δ it would be detectable if we followed the faulty cell (on its own diagonal) by one of the sequences that distinguish states 1 and 3. We observe from the corresponding row that any continuation beginning with α or γ distinguishes these two states from each other and that any continuation beginning with β or δ will not. We conclude that if the initial state is 2 and the fault of type α-δ the corresponding error will propagate if the faulty cell is immediately followed by one giving an output symbol

that is either α or γ and will not propagate if the following cell gives an output which is either β or δ.

The table of Figure 25(e) summarizes the conditions necessary for error propagation. The entries show, for each type of fault, how we must continue the sequence past the faulty cell in order to propagate the error (as a function of the state of the sequence preceding that cell). We have also noted, on the same rows as the state designators, what sequences of symbols preceding the given cell correspond to those states. (In this table a slash mark means "or.")

As an example of the use of this table we observe that a fault of type α-δ will cause error propagation to a remote output boundary of an array when the sequence immediately preceding (to the southwest of) the faulty cell ends in β, or in α preceded by γ or δ, and when the faulty cell is immediately followed (to the northeast) by α or γ. If the sequence proceding the cell is any other than that one, the error will be propagated no matter what the following sequence may be.

It is possible to interpret the table of Figure 25(b) for error propagation from two or more faulty cells on the same diagonal, although we shall not bother to do that here. It is clearly possible, using the table, to determine for <u>any</u> two sequences on that diagonal whether or not they lead to a common sequence on a distant diagonal. When faults occur on more than one diagonal the tables we have derived certainly contain the information requisite to the analysis, but the prospect of actually carrying it out would be a horrendous one.

The reader can verify for himself that the other two tables in Figure 25 (expressing transformations to distant diagonals) <u>also</u> lead to the display given in Figure 25(e), even though the details of how output symbols are associated with the transitions differ markedly from those in the first table. From the method used in the derivation of

these two tables it is apparent that this error-propagation table [Figure 25(e)] is valid for <u>any</u> sequence on the diagonal containing the faulty cell.

For a final step in the analysis of the $\gamma\ \alpha\ \beta\ \beta$ example we recall that Figure 20 shows the sets of sequences that can exist deep in an array. We can now particularize the error propagation analysis to the situation in which the (fault-free) sequences available on the diagonal with the faulty cell are limited to those of the deep-set. We first observe that the symbol δ does not occur in deep-set sequences. This implies that the only possible location for a δ symbol is at a faulty cell. It also tells us that errors of the type δ-a, δ-β, and δ-γ need not be considered. Some typical examples of error propagation analysis for faulty cells deep in the array follow. Figures 20 and 25(e) are vital to this analysis.

For α-δ faults we have determined that in order to propagate errors from an α-δ fault under certain circumstances we must immediately follow the (nominal) α in the faulty cell by a cell with a symbol α or γ. We can see from the deep-set state diagram, however, that α can be immediately followed only by α or γ. Therefore any extension of a valid deep set sequence will propagate the error.

For β-α faults the nominal β in the faulty cell should not be immediately preceded by γ. This restriction is <u>not</u> automatic in the deep-set.

For γ-α faults we should not immediately precede the nominal γ in the faulty cell by α or β unless we immediately follow that same cell by α or γ. A careful consideration of this restriction and the deep-set restrictions taken together tells us that the faulty cell either must be immediately followed by a γ, or must be immediately preceded by γ and immediately followed by β.

Faults of types β-γ, γ-β, and β-δ are obviously propagated regardless of the makeup of the diagonal sequence in which the faulty cell is embedded.

Conditions for propagating errors for the remaining fault types can be obtained in a similar way. For each fault type it turns out that there is some way of propagating the associated errors, even when the faulty cells lie arbitrarily far from both the input and output boundaries of the array.

VI AN ALTERNATIVE APPROACH TO THE ERROR PROPAGATION PROBLEM

We found, in the straightforward approach used in the preceding section, that the number of states required in a state table that described the transformation from a sequence of symbols on one diagonal to the corresponding sequence on another diagonal would, in general, be exponentially related to the separation of the diagonals from each other. In the limit, therefore, the required number of states could not be bounded by any finite number. In this section we shall develop an alternative approach to the problem of determining error propagation from faulty cells. This method presently appears to require only a finite number of states for the associated state tables. In this latter method we separate the propagation problem into two parts.

The possibilities of propagating errors from a faulty cell to the two array output boundaries (to the south and to the east of that cell) are considered separately. Each "half" of the problem is associated with its own state table. The first table tells us how the sequence of symbols on the diagonal containing the faulty cell is mapped onto the sequence of signals on the output boundary to the south. The second table shows how the diagonal sequence is mapped onto the output boundary to the east. We begin the work of this section by deriving the diagonal-to-south-boundary transformation. Our illustrative example (α α γ β) is the one discussed in detail in the first sections of this report.

In Figure 26(a) we show an example of how the diagonal and boundary sequences are related to each other. We assume that there is an "origin" (which, for purposes of discussion, may be taken to be somewhere on the south boundary of the array) from which both sequences start. It is

then clear that the diagonal sequence uniquely determines a corresponding sequence on the boundary. The entries in this example were derived with the aid of the matrix in Figure 2(d).

Nomenclature appropriate to the transformation is explained in Figure 26(b). The initial state is associated with a sequence of (0,1) digits at the west edges of a column of cells in the array. The next-state code corresponds to a sequence of digits on the east edges of that same column of cells. The entries in the state table that is our ultimate goal will show how the prefixed next-state code and the boundary symbol are functionally dependent upon the initial-state code and the diagonal input symbol. For the moment, however, we shall concentrate our attention on how the next-state code (without the prefix) is determined. It will then be an easy matter to prefix that code with the second component of the diagonal input symbol.

The table in Figure 26(c) is obtained directly from the cell logic description of Figure 1. An equivalent representation is shown in Figure 26(d). In this latter form it describes how the sequence of digits in the next state code (taken from north to south) depends upon the corresponding sequence of digits in the code for the initial state. The proper entry node in the diagram is the one which corresponds to the first component of the diagonal input symbol (in the example of the diagram this is "b"). The digits of the code for the initial state determine the path through the diagram, and the identity of the digits of the code for the next state are, in turn, determined by that path. The terminal node of that path corresponds to the boundary symbol at the south end of the column of cells.

Summary statements that describe the state code transformations are:

(1) If the symbol at the top of the column is "a", the symbol at the bottom of the column is also "a" and the next-state code is the same as the initial-state code.

(2) If the symbol at the top of the column is "b" and if the initial state code consists exclusively of 1's (one or more) or is λ (our symbol for <u>a sequence of zero length</u>, not to be confused with a <u>null</u> sequence), the symbol at the bottom of the column is "b" and the next-state code consists, respectively, exclusively of 0's (one or more) or λ. Otherwise (that is, if the initial state code has one or more 0's), the symbol at the bottom of the column is "a" and the next-state code is obtained from the initial-state code by replacing by 0's whatever group of consecutive 1's (if any; not to be preceded by 0's) begin the code.

The transformation which is our goal cannot be a finite-state one unless we find some way to partition all possible state codes into a finite number of disjoint sets. It can be seen from the summary statements above that we must certainly distinguish between codes having no 0's and those having at least one 0 if we are to retain enough information to determine the boundary symbol from the diagonal input symbol. Recall that that diagonal symbol determined the prefix to be attached to the next state code. If the diagonal symbol is [illegible] (shorthand for the pair b1) the prefix is 1 and the symbol at the top of the column is "b." Whether the (prefixed) next-state code will have <u>no</u> 0's or <u>some</u> 0's depends upon whether the initial-state code is λ or has one or more digits, respectively. Consequently, in partitioning the state codes into disjoint sets we must also distinguish between the λ code and the

other codes. The λ code can, of course, exist only at the "origin" from which the diagonal and south boundary sequences both start [see Figure 26(a)].

There are three disjoint sets into which all state codes must be placed:

(1) the λ code,

(2) codes of non-zero length, consisting entirely of 1's, and

(3) codes having one or more 0's.

These three sets are the "states" in the table of Figure 28(b), the state table which was our original goal. The entries of that table have been determined from (a) the definitions of the three sets of state codes above, (b) the fact that the diagonal symbols α, β, γ, and δ contribute the prefix (0, for α and β; or 1, for γ and δ) for the next-state code, (c) the fact that the first components of these symbols (a or b) determine the basic transformations to be applied to the initial-state code, and (d) the summary statements about these transformations listed earlier.

As a typical example of how entries in the state table of Figure 28(b) are derived consider the entry in row 2 and column β. State 2 corresponds to some code consisting entirely of one or more 1's. The β contributes a "b" which determines that the transformation on that code is one which changes all of these 1's to 0's and yields a "$\underline{b}$" symbol (at the bottom of the column of cells). The transformed next-state code, prefixed by the 1 contributed by the β then consists of that 1 followed by one or more 0's; that is, the new code corresponds to state $\underline{3}$. Consequently the entry to be placed at the intersection of row 2 and column β is 3^{b}. The other entries in the table in Figure 28(b) were computed using this same procedure.

This table tells us how an arbitrary sequence of cell output symbols on a diagonal determines the sequence of symbols along the line of cell edges that procedes to the east from the south edge of the first cell on that diagonal. The appropriate "starting state" for such transformations is state 1 because the zero-length (λ) code is a member of the set of codes associated with that state. We note from the state table that that state can occur only as the starting state. We can also observe that the only way of staying in state 2 is to follow it only by γ symbols. If any of the other diagonal symbols occurs state 3 will be entered and, once entered, cannot be left. Finally, we note that when the diagonal sequence is one associated with state 3, the resulting boundary symbol is "a," independent of what further sequence of symbols occurs on the diagonal.

The state table [Figure 28(b)] can be used to determine whether or not errors from a faulty cell will propagate to the south boundary of the array. If the state of the diagonal sequence preceding the faulty cell is "1" (this can happen only when the faulty cell is on the array boundary) we note that faults of types α-β, α-δ, β-α, β-γ, γ-β, and γ-δ involves changes that will cause immediate changes of the boundary symbol, because the two boundary symbols associated with these faults are different. (Because no cell for this example can have a nominal output that is δ we need not concern ourselves with faults of types δ-α, δ-β, or δ-γ.)

When the diagonal state is "1" the other possible types of faults (α-γ, γ-α, and β-δ) will cause symbol changes on the boundary for exactly those continuations of the diagonal sequence that allow us to distinguish between states 2 and 3. By examining the corresponding two rows of the table we see that an α symbol is not an appropriate continuation if we wish to continue to propagate the error, because in the α column both next-states and boundary symbols are the same. A β continuation, on the

other hand, always causes differing boundary symbols. When the continuation of the diagonal begins with γ we see that the boundary symbol is "a" regardless of whether we were in state 2 or state 3. If the continuation symbol is γ the next state on the diagonal is again 2 or 3 depending upon whether or not the fault is present. We conclude that when a cell with a fault of type α-γ, γ-α, or β-δ is on the south boundary of the array the fault will be detected only if the diagonal sequence is continued with β, or with one or more γ's followed by a β. From the description of the deep-set [Figure 28(a) as been copied from Figure 12(a)] we can conclude that an all-γ diagonal sequence can always be continued either with β, or with one or more γ's followed by β.

If the faulty cell is preceded on its diagonal by an all-γ sequence the corresponding state will be state 2. In that case it is obvious from the table that faults of types α-β, α-δ, β-α, β-γ, γ-β, and γ-δ will cause errors which propagate directly to the south boundary. If the cell has a fault of type α-γ or γ-α that cell should be followed on the diagonal by some continuation which allows us to distinguish between states 2 and 3. (We considered how to do this in the preceding paragraph.) If the cell has a fault of type β-δ that fault will not be detectable regardless of how we continue the diagonal sequence.

If the faulty cell is preceded on its diagonal by any sequence associated with state 3 it is apparent from the table of Figure 28(b) that no fault type will be detectable.

We have demonstrated for this example that a faulty cell deep in the array cannot propagate errors to the south boundary of the array unless it is preceded on its diagonal only by γ's. If it is preceded by such a sequence then all faults except the β-δ type can be detected if the diagonal sequence is continued past the faulty cell in ways that are possible even when the diagonal is deep in the array.

The development of the transformation from the sequence of diagonal symbols to the sequence of symbols on the east boundary of the array (see Figure 27) parallels the one we have just completed. For this new transformation we think of the diagonal sequence as proceeding in the "reverse" direction (to the southwest) and the east boundary signals as proceeding to the south. The altered description of cell logic is equivalent to the earlier one and leads directly to the diagram of Figure 27(d).

The summary statements that describe the state-code transformations are:

(0) If the symbol at the left of the row [of cells in Figure 27(b)] is 0, the symbol at the right of the row is also 0 and the next-state code contains only a's (or is λ).

(1) If the symbol at the left of the row is 1 and if the initial-state code consists exclusively of a's (one or more) or is λ, the symbol at the right of the row is 1 and the state code is unchanged. Otherwise (that is, if the initial-state code contains b's) the symbol at the right of the row is 0, only the first consecutive subsequence of b's is retained in the state code, and all other b's are replaced by a's.

There are only two disjoint sets into which all state codes need be placed:

(1) codes containing no b's (includes λ) and

(2) codes containing at least one b.

The derivation of the state table of Figure 28(d) is left as an exercise for the reader. [This derivation parallels the one used earlier in obtaining Figure 28(b).] The deep-set for the reversed deep-diagonal sets is given in Figure 28(c) and is derived directly from Figure 28(a).

The reader may verify for himself the validity of the following statements that were obtained from interpreting the table in Figure 28(d). Note especially that in these statements when we refer to the sequence "preceding" a faulty cell we mean the sequence to the northeast of that cell and when we refer to the sequence "following" the faulty cell we mean the sequence to the southwest of that cell.

As a preface we observe that states 1 and 2 of the transducer can be distinguished only when followed by γ or δ (although, as we have mentioned earlier, δ is not possible as the output symbol from a non-faulty cell). We observe also that regardless of the past history of the diagonal sequence both state 1 and state 2 are accessible by appropriately continuing that sequence. State 1 is reached whenever the last symbol of the preceding diagonal sequence is α, or is γ preceded by α. Any other sequence results in state 2.

When the sequence preceding the faulty cell results in state 1, errors from faults of types α-γ, α-δ, β-γ, β-δ, γ-α, and γ-β are always propagated. Those from types α-β, β-α, and γ-δ are propagated when the faulty cell is immediately followed by γ. When the sequence preceding the faulty cell results in state 2, errors from faults of types α-β, α-γ, α-δ, β-α, and γ-α are propagated only if the cell is immediately followed by γ. The errors from all other faults are not propagated to the east boundary of the array.

In our example of this section it has turned out that errors could be propagated from all fault types either to the south boundary or to the east boundary of the array (or to both). In general certain errors may propagate to neither boundary, or all errors that can be propagated may be propagated only to one or the other of these boundaries. The possibilities are numerous and we have discovered no general method that will allow us to determine directly from the cell logic what these

possibilities are. In the cases studied during the course of this research it has always been possible to derive finite-state transformations from the diagonal sequence to each of the (south and east) output boundaries of the array. We nevertheless hesitate to conclude that this is always possible and, admittedly, the derivation of these transformations can be somewhat involved. On the other hand the research reported in the preceding section demonstrates that the obvious diagonal-to-diagonal transformation cannot, in general, result in finite state tables, regrettable as this may be. Consequently this new approach constitutes a decided improvement upon that one.

VII MORE EXAMPLES OF THE STATE-TABLE APPROACH TO ERROR PROPAGATION TO ARRAY BOUNDARIES

In this section we shall show two more examples of the derivation of the types of state tables we illustrated in the last section. These were chosen so as to show the range of difficulties encountered during the period of this research. We shall also illustrate how the pair of error-propagation tables (showing how errors can be propagated from diagonals on which we are free to choose any sequence of symbols we desire) and the pair of tables that describe the deep-sets can be combined. The combined tables tell us specifically how errors from faulty cells that may be arbitrarily deep in the array can be propagated.

The first of these examples is the $\alpha\ \beta\ \beta\ \gamma$, or "half-adder," example for which the deep-sets have already been considered (see Figures 21 and 22). In Figure 29 are given the critical steps in the derivation of the table showing the transformation from the diagonal sequence to the sequence on the south boundary of the array. The transformation that tells us how the code for an initial state is changed to the (unprefixed) code for the next state follows directly from the description of the cell logic. We observe that basically this transformation reduces the number of 1's in the code by half by replacing every other 1 by a 0.

A more detailed analysis shows that if the number of 1's in the initial state code is k, then the number of 1's in the (unprefixed) next-state code is

$k/2$ if k is even and we start at the "a" node,
$(k-1)/2$ if k is odd and we start at the "a" node,
$k/2$ if k is even and we start at the "b" node, and
$(k+1)/2$ if k is odd and we start at the "b" node.

Preceding page blank

Furthermore the resulting south-boundary symbol [see Figure 26(b)] is a, b, b, or a, respectively, for these four cases.

The corresponding rules when we take into consideration the prefix supplied by the diagonal symbol are summarized in Figure 29(c). For instance if k is odd and the diagonal symbol is δ (=bl) w[illegible]d the prefix 1 to the code that results from starting at the "b" node. The result is a total of $1 + (k+1)/2 = (k+3)/2$ 1's.

Not all values of k can occur in states that are accessible in the desired transformation. This transformation is shown in Figure 29(d). In that table the label for the state corresponds to the value of k. The starting state (code λ) has no 1's and therefore can lead only to itself or to a code having one 1. State 1 can lead to states 0, 1, and 2. State 2 can lead only to states 1 and 2. Consequently only these three states are accessible from the starting state. The superscripts show what the south-boundary symbol is.

In this case we can easily see that all types of faults correspond to errors that propagate to the south boundary. State pairs 0, 1, and 1, 2 are always immediately distinguishable, regardless of what diagonal symbol occurs immediately following the faulty cell. The only state from which the state pair 0, 2 might possibly have to be considered is state 1, for fault-type α-δ. However, that fault can be seen to give different south-boundary symbols and on that latter basis we conclude that it can be propagated.

Because it is clear from the discussion of this half-adder example that all errors propagate to the south boundary we need not consider the transformation from the (reversed) diagonal sequence to the east-boundary sequence.

In the other example ($\alpha\ \gamma\ \beta\ \gamma$) of this section we shall now see that state diagrams that show exactly how the diagonal sequence is mapped onto the south and east boundaries would require an <u>infinite</u> number of states. Nevertheless we shall demonstrate in a <u>finite</u> table how to express the necessary distinctions among state codes.

In Figure 30(b) is given the transformation between initial-state codes and the resulting (unprefixed) next-state codes. It is easy to see that if an arbitrary sequence of digits is transformed the resulting sequence cannot contain two or more consecutive 1's. (After such a transformed code is prefixed by 1 it may of course begin with two consecutive 1's.) As a good first approximation the effect of the transformation on codes with such constraints is to shift the digits of the code to the right, by first attaching a 0 or 1 to the left end of the code (depending upon whether we begin with node "a" or node "b," respectively) and then deleting the rightmost digit. For instance if the sequence 1 1 0 1 0 0 1 0 1 0 0 0 1 were transformed beginning at node "b" the resulting code would be 1 0 1 0 1 0 0 1 0 1 0 0 0.

We show in Figure 30(c) the details of how the first few digits of the code for the initial state are transformed into the first few digits of the prefixed next-state code. As usual, the right component of the diagonal symbol is the prefix. The left component of the diagonal determines the node from which the path is begun in the state diagram of Figure 30(b).

The south-boundary symbol is determined by the identity of the rightmost digit of the state code. If that digit is 0 the resulting symbol is "a;" if that digit is 1 (necessarily preceded by a 0 if the code has three or more digits) the resulting symbol is "b." Therefore the fact that the 1's in a code are shifted as we progress from one code to the next (that is, shifted to the south in the array as we progress from one column of cells to the one to the east) guarantees

that a 1 in a state code will eventually determine whether some south-boundary symbol to the southeast will be "a" or "b." It is therefore impossible to associate with the entries of the table in Figure 30(c) a specific "a" or "b" symbol because each of the four rows corresponds to a wide variety of arbitrarily long codes. We do know, however, that if two codes differ in their fourth digits then that difference will ultimately have a direct effect at the south boundary of the array. Furthermore, we note for emphasis that these four classes of codes are not equivalence classes, for each class is associated with an infinite set of codes, most pairs of which we know to be distinguishable.

From the analysis above we can determine under what circumstances changing a symbol on the diagonal will ultimately cause a corresponding change on the boundary. That is, we tell when an error from a fault will lead to a state which is "ultimately distinguishable" from the state that would have resulted had the cell continued to produce its nominal output symbol.

As an example of the way this table can be interpreted we assume that the diagonal sequence preceding a faulty cell corresponds to a code 0 1 ··· . If there is a β-δ fault at that cell the corresponding two codes are 0 1 0 1 ··· and 1 1 0 1 ··· . The third and fourth digits (and all succeeding digits) of these codes are the same and therefore we cannot guarantee that they will lead ultimately to different boundary symbols. However, by observing the first two digits of these codes and comparing the rows of the table that are labelled 0 1 ··· and 1 1 ··· we see that the fault in the cell will be propagated if the next diagonal symbol is α or γ and will not be propagated if the next symbol is β or δ.

A similar interpretation of the other entries in the table reveals that faults of types α-β, α-δ, β-α, β-γ, or γ-β will always be propagated,

as will faults of type γ-δ if these latter faults occur on the diagonal at locations corresponding to state codes 1 0 ... or 1 1 ... . [These states correspond to a symbol γ (= a$\underline{1}$) or δ (= b$\underline{1}$) in the preceding cell. The second of these is not a possibility if there is only one faulty cell.] Faults of types α-γ, β-δ, or γ-α will always be propagated if they are followed on the diagonal by α or γ. A fault of type γ-δ will be propagated if it occurs on the diagonal at a location corresponding to a state encoded 0 0 ··· or 0 1 ··· [that is, the fault should be preceded on the diagonal by a cell with output α(= a$\underline{0}$) or β(= b$\underline{0}$)] and if the faulty cell is followed by β or δ.

The transformation from (reversed) diagonal sequences to east-boundary sequences involves issues much like those considered above. In Figure 31 the essential steps in this part of the analysis are given. In this case the code transformation applied to an arbitrary code (as a first approximation) replaces all but the final b in each group of consecutive b's by a's and shifts the remaining b's by one unit. Therefore a transformed code has (except possibly in its first two digits) no two consecutive b's. The details of what happens to the first few digits of codes is shown in Figure 31(c). It can be observed that an "a" or "b" as the final digit of a code directly determines whether the boundary symbol produced is 0 or 1, respectively. If codes differ in their third digits it is certain that the corresponding states will be ultimately distinguishable by this boundary symbol.

As an exercise the reader should determine for himself the truth of the following statements. (Recall once more that, for the transformation, "preceding" means from the northeast and "following" means to the southwest.) Faults of types α-β, β-α, β-γ, and γ-β are always propagated. Faults of types α-δ and γ-δ are always propagated if they occur on the diagonal at a location corresponding to a state encoded by ba... or bb... [that is, there must be a cell immediately to the northeast giving

as its output β (= $\underline{b}0$) or δ (= $\underline{b}1$), the latter case an impossibility if there is only one faulty cell]. A fault of type β-δ will be propagated if it occurs at a location corresponding to codes aa... or ab... [that is, there must be a cell immediately to the northeast giving an α (= $\underline{a}0$) or γ (= $\underline{a}1$) as its output]. The errors from all other faults will not be propagated to the east boundary.

The tables of Figure 30(c) and Figure 31(c) show how errors are propagated to a remote south boundary and to a remote east boundary from a faulty cell on a diagonal on which the sequence of symbols can be chosen with no constraints whatsoever. If we wish to restrict the error propagation analysis to cells that lie deep in the array, and therefore on diagonals along which only sequences from the deep-set can exist, it is convenient to introduce composite tables (one for each output boundary) that contain all the information needed. A significant advantage of the state-table approach to the excitation and propagation problems is that such composite tables are possible.

The deep-set for the α γ β γ example was shown in Figure 18 and is repeated in Figure 32(a) and 32(b). The large table in that figure has as its states products of states from this deep-set table and states from the error-propagation table of Figure 30(c). The entries within the table have two components, each of which is computed independently from its corresponding table. Since the last two digits of each entry are useful in determining the circumstances under which two states are ultimately distinguishable, they serve somewhat the same role as the outputs would in a conventional description of a finite-state transformation. These digits have therefore been shown as superscripts.

This composite table is especially useful in determining whether or not a faulty cell can be followed (to the northeast) by some continuation of the diagonal sequence that would, on the one hand, propagate

the error to the south boundary and that also, on the other hand, is one of those available from the deep-set. We enter the table at one of the rows P00, P01, or Q10 since these can be seen to correspond to the three state-groups that are accessible if no faults exist on the diagonal. The particular row we enter is determined by the composition of the deep-diagonal sequence we assume has preceded the faulty cell. For purposes of illustrating the procedure we shall assume that this preceding sequence corresponds to P01.

If the faulty cell is the next one on the diagonal and it has a nominal output of γ the corresponding new composite state is $Q10^{01}$. If that cell had a γ-δ error the nominal deep-set sequence would of course remain unchanged (that is, its contribution to the composite state would still be Q) but the contribution to the composite state from the error propagation table would be changed (from 10^{01}) to 11^{01}. In other words the γ-δ fault at the specified location on the diagonal corresponds to the two entries $Q10^{01}$ and $(Q)11^{01}$. Because the superscripts of these two entries are the same we conclude that in order to determine whether or not the error can be propagated we must next examine corresponding pairs of entries from the rows Q10 and Q11. The entries from these rows indicate that indistinguishable states will be produced, whichever symbol (α or γ) is produced by the next cell on the diagonal. If we <u>were</u> able to follow the faulty cell by a β the superscripts of the entries -01^{00} and -01^{01} tell us that the resulting states <u>would</u> be ultimately distinguishable. Therefore the γ-δ fault cannot propagate errors to the south boundary if the continuation sequence on the diagonal is restricted to one from the deep-set, although it can be propagated if that restriction does not apply.

The example cited above indicates that even with the aid of the table we have derived, the issue of detectability of faults is still a somewhat complex one. Whether or not errors will be propagated from

a faulty cell deep in the array is a function of the particular deep-set sequence that exists on the diagonal containing that faulty cell. The table does, nevertheless, gather together in one place just that information necessary to answer such questions. A detailed analysis of this present table shows that the only fault that might not propagate errors to the south boundary is the γ-δ type. The error will not be propagated if the sequence preceding the faulty cell corresponds to P00 or P01, but will be propagated if that sequence corresponds to Q10.

In Figure 33 we show the derivation of the composite table showing the conditions under which errors can be propagated from faulty cells deep in the array to the east boundary of the array. From it we can determine, for example, that if the sequence preceding (from the north-east) a cell with a β-γ fault is in the class Raa the result without the fault is Sba^{a} and with the fault is $(S)ab^{a}$. By referring to corresponding pairs of entries on rows Sba and Sab we note that either α or β (both are allowable in deep-set sequences) causes ultimately distinguishable states to follow. Consequently the fault is detected.

A complete analysis of the composite table reveals that faults of types α-γ and γ-α cannot be propagated (to the east boundary), and fault type β-δ can not be propagated if the sequence preceding it is associated with Sba.

In the final section we summarize our research efforts and suggest what future research in this problem area might be of value.

VIII SUMMARY AND SUGGESTIONS FOR FUTURE STUDY

New methods have been found to meet the objectives announced in the Introduction. We have also tried at each stage to indicate the difficulties involved in both the excitation and propagation problems, showing both elementary examples and examples of more inherent complexity.

Because an objective of this report has been to discuss all procedures in the detail necessary for the reader to understand them fully and reproduce them himself there has been created, perhaps, the impressions that these procedures are somewhat complex. With a very little effort however, the reader can see for himself that each stage of development is quite straightforward and necessary, given the inherent complexity of the problems considered and the generality of the solutions which have been offered.

For the reader who feels that perhaps direct simulation of a cellular array would produce the same results as the procedures developed in this report we offer the reminder that the number of combinations of possible input signals to an array is an exponential function of the dimensions of the array. Our procedures are valid for entire systems of arrays; that is, they are valid for cellular arrays of arbitrary size, however large, and depend only upon the logical structure of the individual cells that are used as the building blocks of the array.

We believe that the main contributions resulting from this study are

(1) the "deep-set" concept, which includes within it the problem of considering all possible tesselations,

(2) the separation of the error propagation problem into two components, a separation that seems necessary if the propagation problem is to be expressed in finite tables,

(3) a recognition of the correspondence between the concept of state non-equivalence and error propagatability, and

(4) an introduction of the concept of "ultimate distinguishability" that allows us to discuss some infinite state transformations in finite-state terms.

It is, nevertheless, not yet possible for us to claim that a completely general algorithm exists for the diagnosis of faults in two-dimensional arrays with unilateral signal flow. Two major logical gaps must be bridged before that claim would be justified. The first would be filled if we had a definitive algorithm for the determination of the deep-sets. It is obvious that each member of the sequence of k-level sets (whose limit is the deep-set) can be derived from the preceding member and that each requires only a finite number of states. The limit is apparent when it is reached for a finite value of k. In those many cases studied in which the limit existed only as $k \to \infty$, it was always possible to determine (by an examination of the state diagrams for the sequence of sets) what that limit was. An algorithm is needed that would lead us directly to the limit set from a description of the cell logic. We conjecture that the limit always exists and that its description requires only a finite number of states, but a proof of the conjecture is certainly desirable.

The second logical gap that should be bridged in order to obtain a general algorithm for diagnosis of faults is associated with the two diagonal-to-boundary transformations used in analyzing the problem of error propagation. We know that these transformations can require an

infinite number of states. It is our conjecture that in these cases a reduction to a finite-state description is possible (using the "ultimately distinguishable" concept) but the conjecture should be proved or disproved in general.

Now that the procedural tools developed here are available, a study that relates the logical properties of a cell to the characteristics of the various transformations and sets developed here is indicated. For instance, what properties of a cell indicate that the deep-set will be found only as a limit, or that the diagonal-to-output-boundary transformation will require an infinite number of states?

REFERENCES

1. W. H. Kautz, "Testing for Faults in Combinational Cellular Logic Arrays," Conference Record of 8th Annual Symposium on Switching and Automata Theory, IEEE Publication 16C-40, pp. 161-174 (1967).

2. P. R. Mennon and A. D. Friedman, "Fault Detection in Iterative Logic Arrays," IEEE Transactions on Computers, Vol. C-20, No. 5, pp. 524-535 (1971).

3. F. C. Hennie, Finite-State Models for Logical Machines (John Wiley, New York, N.Y. 1968).

Preceding page blank

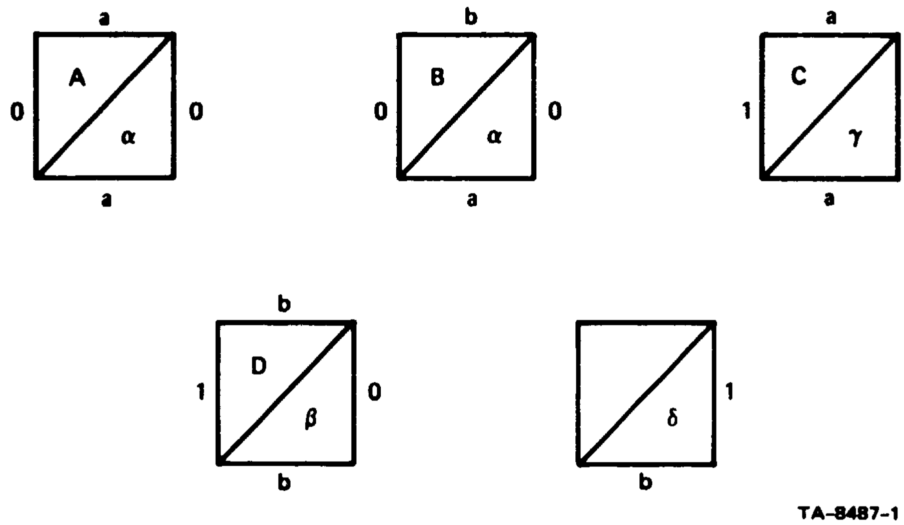


FIGURE 1 AN EXAMPLE OF CELL LOGIC

Preceding page blank

WEST CELL: \ NORTH CELL:	A (a)	B (a)	C (a)	D (b)
A (0)	A	A	A	B
B (0)	A	A	A	B
C (1)	C	C	C	D
D (0)	A	A	A	B

(a)

WEST CELL: \ NORTH CELL:	α (a)	β (b)	γ (a)	δ (b)
α (0)	A	B	A	B
β (0)	A	B	A	B
γ (1)	C	D	C	D
δ (1)	C	D	C	D

(b)

WEST CELL: \ NORTH CELL:	A (a)	B (a)	C (a)	D (b)
A (0)	α	α	α	α
B (0)	α	α	α	α
C (1)	γ	γ	γ	β
D (0)	α	α	α	α

(c)

WEST CELL \ NORTH CELL	α (a)	β (b)	γ (a)	δ (b)
α (0)	α	α	α	α
β (0)	α	α	α	α
γ (1)	γ	β	γ	β
δ (1)	γ	β	γ	β

(d)

TA 8487 2

FIGURE 2 MATRICES SHOWING, FOR THE EXAMPLE OF FIGURE 1, THE DEPENDENCE OF THE STATE OF A CELL UPON THE STATES OF THE CELLS TO ITS WEST AND NORTH

	a	a	a	a	a	a	b	a	a	a	a	a	
1	C	C	C	C	C	C	D	A	A	A	A	A	0
1	C	C	C	C	C	C	D	A	A	A	A	A	0
1	C	C	C	C	C	C	D	A	A	A	A	A	0
1	C	C	C	C_A	C_A	C_A	D_B	A	A	A	A	A	0
1	C	C	C	C	C	C	D_C	A_C	A_C	A_C	A_C	A_C	$0_{(1)}$
1	C	C	C	C	C	C	D_C	A_C	A_C	A_C	A_C	A_C	$0_{(1)}$
0	A	A	A	A	A	A	B_A	A	A	A	A	A	0
0	A	A	A	A	A	A	A	A	A	A	A	A	0
1	C	C	C	C	C	C	C	C	C	C	C	C	1
0	A	A	A	A	A	A	A	A	A	A	A	A	0
1	C	C	C	C	C	C	C	C	C	C	C	C	1
1	C	C	C	C	C	C	C	C	C	C	C	C	1
0	A	A	A	A	A	A	A	A	A	A	A	A	0
	a	a	a	a	a	a	a	a	a	a	a	a	

TA-8487-3

FIGURE 3 AN ARRAY OF CELLS WITH STATES ASSOCIATED WITH CELL INPUTS

	a	a	a	a	a	a	b	a	a	a	a	a	
1	γ	γ	γ	γ	γ	γ	β	α	α	α	α	α	0
1	γ	γ	γ	γ	γ	γ	β	α	α	α	α	α	0
1	γ	γ	γ	γ	γ	γ	β	α	α	α	α	α	0
1	γ	γ	γ_α	γ_α	γ_α	γ_α	β_α	α	α	α	α	α	0
1	γ	γ	γ	γ	γ	γ	β_γ	α_γ	α_γ	α_γ	α_γ	α_γ	$0_{(1)}$
1	γ	γ	γ	γ	γ	γ	β_γ	α_γ	α_γ	α_γ	α_γ	α_γ	$0_{(1)}$
0	α	α	α	α	α	α	α	α	α	α	α	α	0
0	α	α	α	α	α	α	α	α	α	α	α	α	0
1	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	1
0	α	α	α	α	α	α	α	α	α	α	α	α	0
1	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	1
1	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	γ	1
0	α	α	α	α	α	α	α	α	α	α	α	α	0
	a	a	a	a	a	a	a	a	a	a	a	a	

TA-8487-4

FIGURE 4 AN ARRAY OF CELLS WITH STATES ASSOCIATED WITH CELL OUTPUTS

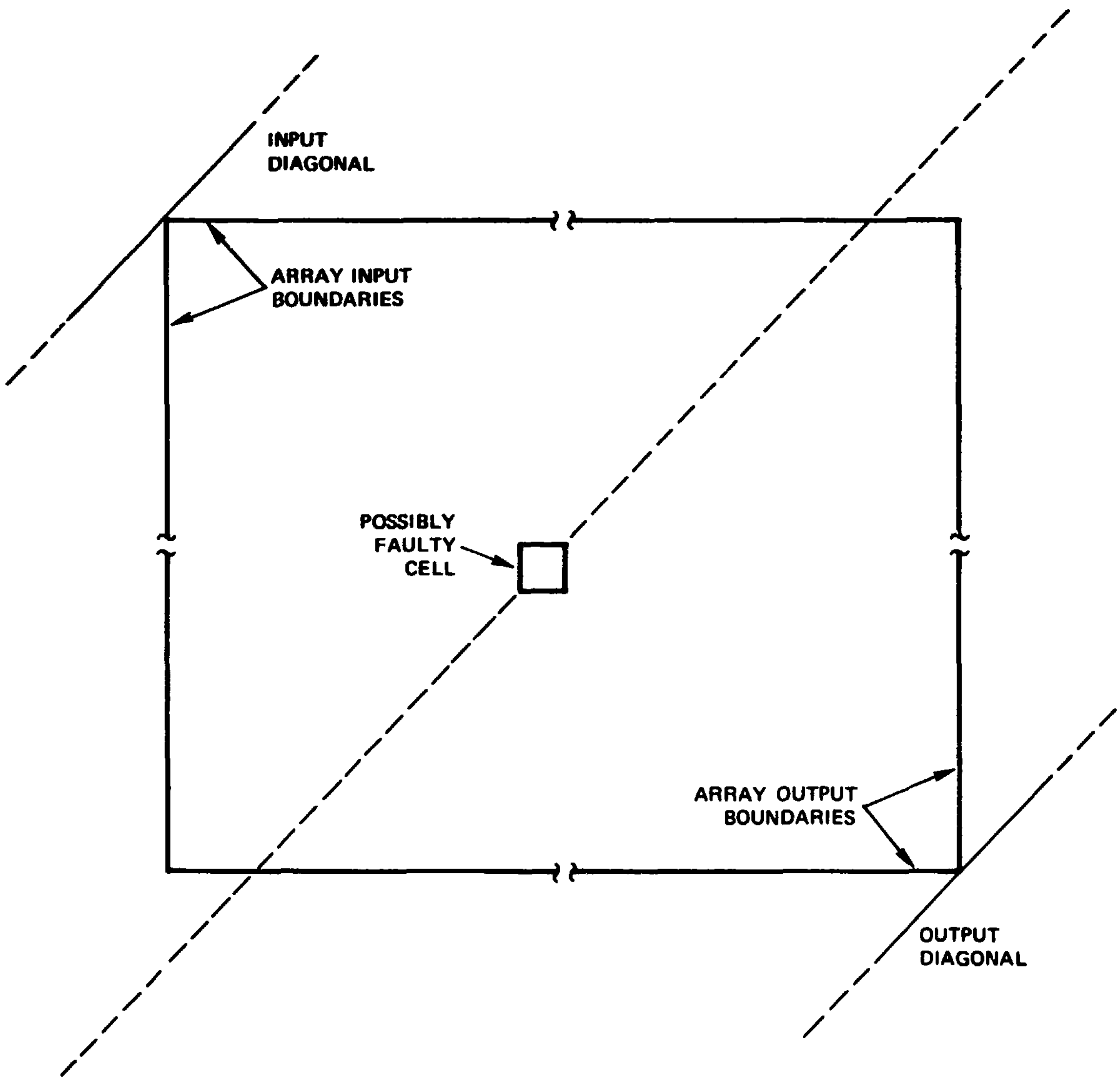
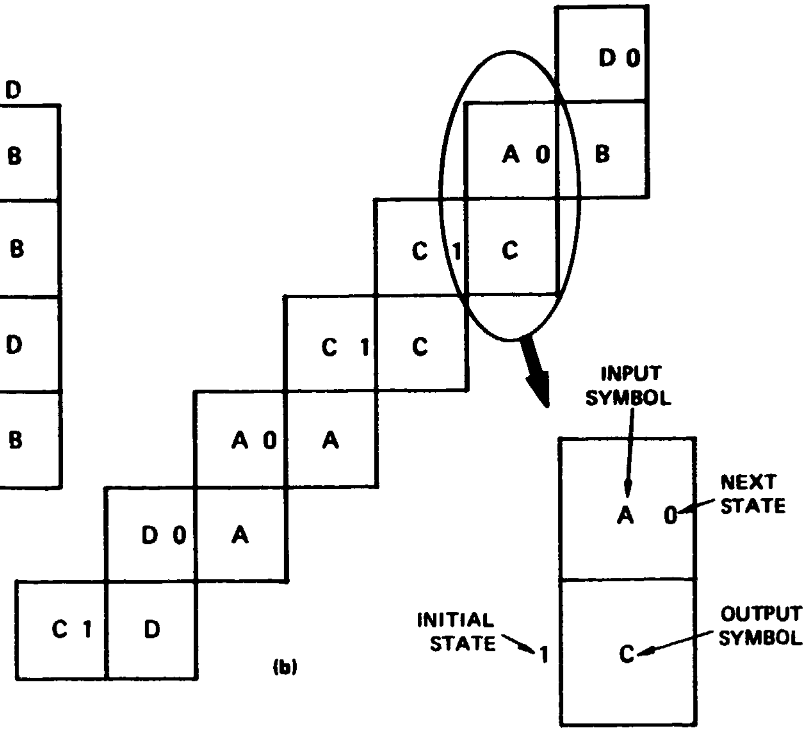


FIGURE 5 DIAGRAM SHOWING JUSTIFICATION FOR CONSIDERING DIAGONAL-TO-DIAGONAL SIGNAL FLOW

	A	B	C	D
A (0)	A	A	A	B
B (0)	A	A	A	B
C (1)	C	C	C	D
D (0)	A	A	A	B

(a)



(c)

INPUT:

	A	B	C	D
STATE: 0	0^A	0^A	1^A	0^B
1	0^C	0^C	1^C	0^D

(d) THE DIAGONAL-TO-DIAGONAL TRANSDUCER, T

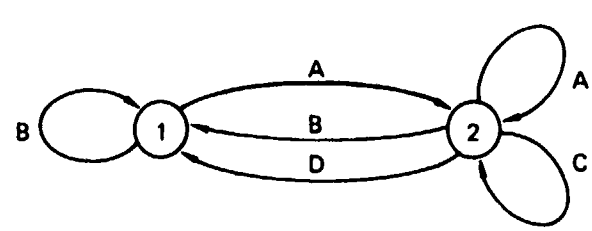
OUTPUT:

	A	B	C	D	
STATE 0	01	0	–	–	✓
1	–	–	01	0	
01	01	0	01	0	✓

(e) THE OUTPUT TABLE

	A	B	C	D
STATE: 1	2	1	–	–
2	2	1	2	1

(f) THE 1-LEVEL SET



(g) THE 1-LEVEL SET

TA-8487-6

FIGURE 6 DERIVATION OF THE 1-LEVEL DIAGONAL SET

INPUT:

STATE	A	B	C	D	
10	20^{A}	10^{A}	–	–	√
11	20^{C}	10^{C}	–	–	
20	20^{A}	10^{A}	21^{A}	10^{B}	√
21	20^{C}	10^{C}	21^{C}	10^{D}	√

(a) THE TRANSFORMATION FROM THE 1-LEVEL TO THE 2-LEVEL

INPUT:

STATE	A	B	C	D
1	2^{A}	1^{A}	–	–
2	2^{A}	1^{A}	3^{A}	1^{B}
3	2^{C}	1^{C}	3^{C}	1^{D}

(b) THE SAME TRANSFORMATION, RELABELLED

OUTPUT:

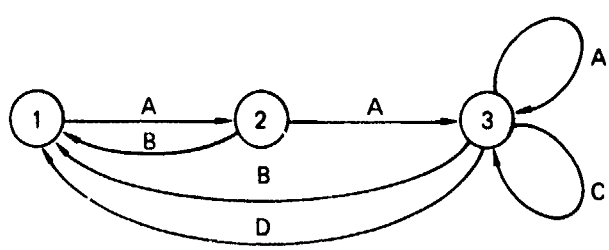
STATE	A	B	C	D	
1	12	–	–	–	√
2	123	1	–	–	
3	–	–	123	1	
12	123	1	–	–	√
123	123	1	123	1	√

(c) THE OUTPUT TABLE

OUTPUT:

STATE	A	B	C	D
1	2	–	–	–
2	3	1	–	–
3	3	1	3	1

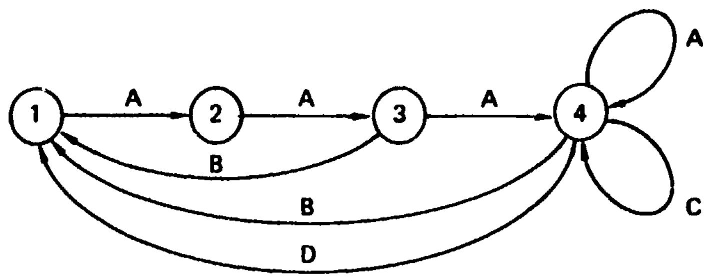
(d) THE 2-LEVEL SET



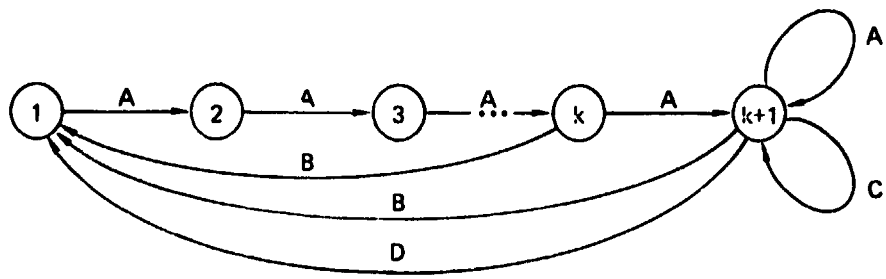
(e) THE 2-LEVEL SET

TA-8487-7

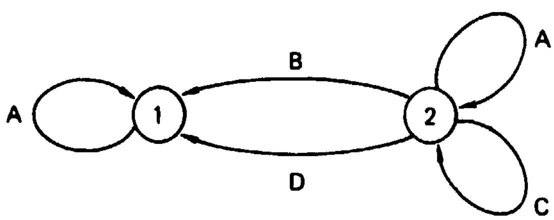
FIGURE 7 DERIVATION OF THE 2-LEVEL SET



(a) THE 3-LEVEL SET



(b) THE k-LEVEL SET (k ≥ 4)



OUTPUT:

STATE	A	B	C	D
1	1	–	–	–
2	2	1	2	1

(c) THE LIMITING (OR "DEEP") SET

TA-8487-8

FIGURE 8 DERIVATION OF THE DEEP-SET

STATE \ INPUT	A	B	C	D	
10	10^A	–	–	–	√
11	10^C	–	–	–	
20	20^A	10^A	21^A	10^B	√
21	20^C	10^C	21^C	10^D	√

(a) THE TRANSFORMATION FROM ONE DEEP DIAGONAL TO THE NEXT DIAGONAL

STATE \ INPUT	A	B	C	D
1	1^A	–	–	–
2	2^A	1^A	3^A	1^B
3	2^C	1^C	3^C	1^D

(b) THE SAME TRANSFORMATION RELABELLED

STATE \ OUTPUT	A	B	C	D	
1	1	–	–	–	√
2	123	1	–	–	
3	–	–	123	–	
123	123	1	123	1	√

(c) THE OUTPUT TABLE

STATE \ OUTPUT	A	B	C	D
1	1	–	–	–
2	2	1	2	1

(d) THE DEEP-SET

TA-8487-9

FIGURE 9 VERIFICATION OF THE DEEP-SET

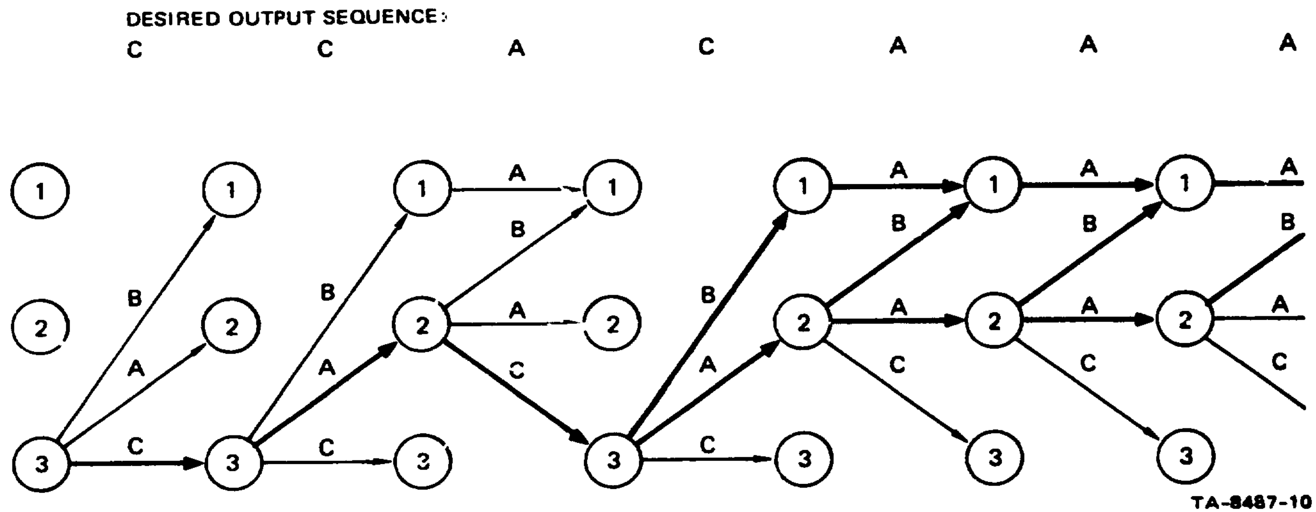


FIGURE 10 POSSIBLE INPUT SEQUENCES WHICH LEAD TO A GIVEN OUTPUT SEQUENCE

INPUT:

	α	β	γ	δ
0	0^{α}	0^{α}	1^{α}	1^{α}
1	0^{γ}	0^{β}	1^{γ}	1^{β}

(a) THE DIAGONAL-TO-DIAGONAL TRANSDUCER, τ

OUTPUT:

	α	β	γ	δ	
0	01	01	–	–	
1	–	01	01	–	
01	01	01	01	–	√

(b) THE OUTPUT TABLE

INPUT:

	α	β	γ	δ
0	0^{α}	0^{α}	1^{α}	–
1	0^{γ}	0^{β}	1^{γ}	–

(c) THE TRANSFORMATION FROM THE 1-LEVEL TO THE 2-LEVEL

OUTPUT

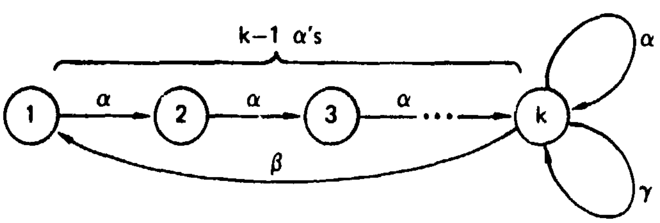
	α	β	γ	δ	
0	01	–	–	–	√
1	–	0	01	–	
01	01	0	01	–	√

(d) THE OUTPUT TABLE

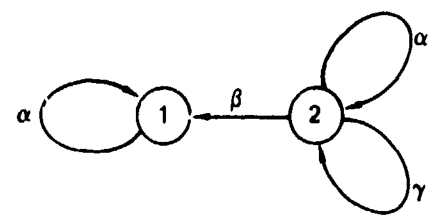
OUTPUT

	α	β	γ	δ
1	2	–	–	–
2	2	1	2	–

(e) THE 2-LEVEL SET



(f) THE k-LEVEL SET



(g) THE DEEP-SET

TA-8487-11

FIGURE 11 DERIVATION OF THE DEEP-SET IN TERMS OF THE CELL OUTPUT SYMBOLS

	α	β	γ	δ
1	1	–	–	–
2	2	1	2	–

(a) THE DEEP-SET

INPUT:

	α	β	γ	δ	
10	10^{α}	–	–	–	√
11	10^{γ}	–	–	–	
20	20^{α}	10^{α}	21^{α}	–	√
21	20^{γ}	10^{β}	21^{γ}	–	√

(b) THE TRANSFORMATION FROM ONE DEEP DIAGONAL TO THE NEXT DIAGONAL

INPUT:

	α	β	γ	δ
1	1^{α}	–	–	–
2	2^{α}	1^{α}	3^{α}	–
3	2^{γ}	1^{β}	3^{γ}	–

(c) THE SAME TRANSFORMATION, RELABELLED

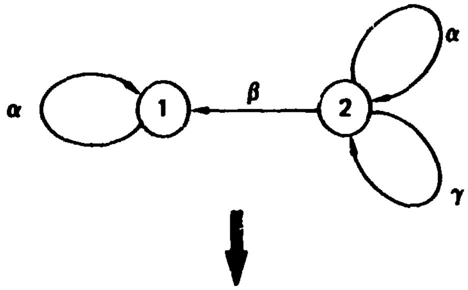
OUTPUT:

	α	β	γ	δ	
1	1	–	–	–	√
2	123	–	–	–	
3	–	1	23	–	
23	123	1	23	–	√
123	123	1	23	–	√

(d) THE OUTPUT TABLE

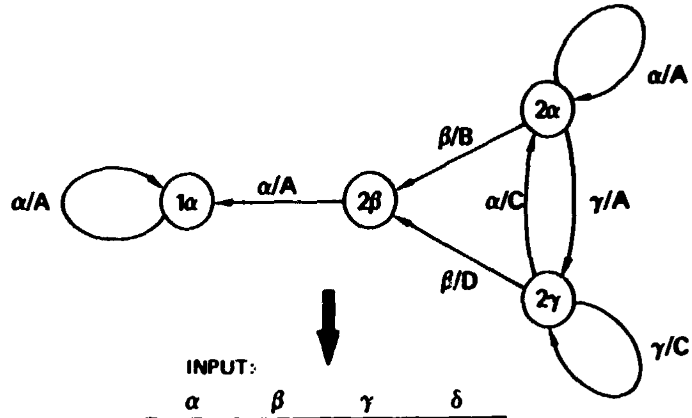
TA-8487-12

FIGURE 12 VERIFICATION OF THE DEEP-SET



	α	β	γ	δ
1	1	–	–	–
2	2	1	2	–

(a) THE DEEP-SET WITH CELL OUTPUT SYMBOLS



INPUT:

	α	β	γ	δ	
1α	$1\alpha^{A}$	–	–	–	(1)
2α	$2\alpha^{A}$	$2\beta^{B}$	$2\gamma^{A}$	–	(2)
2β	$1\alpha^{A}$	–	–	–	(1)
2γ	$2\alpha^{C}$	$2\beta^{D}$	$2\gamma^{C}$	–	(3)

(b) TRANSFORMATION FROM CELL OUTPUT SYMBOLS ON ONE DIAGONAL TO CELL INPUT SYMBOLS ON THE NEXT

INPUT:

	α	β	γ	δ
1	1^{A}	–	–	–
2	2^{A}	1^{B}	3^{A}	
3	2^{C}	1^{D}	3^{C}	–

(c) TRANSFORMATION, RELABELLED

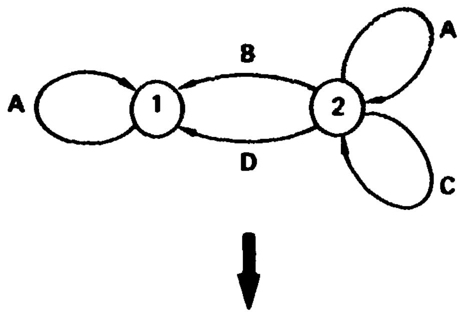
OUTPUT:

	A	B	C	D	
1	1	–	–	–	√
2	23	1	–	–	
3	–	–	23	1	
23	23	1	23	1	√

(d) THE OUTPUT SET

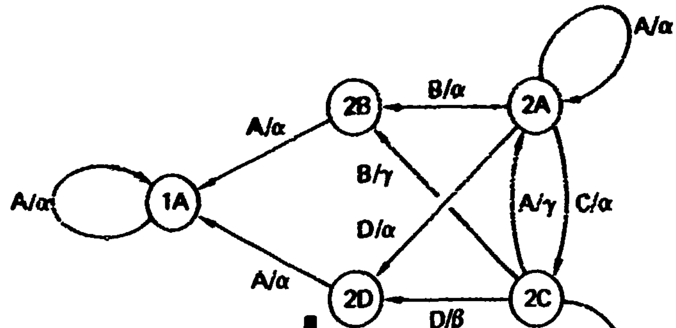
TA-8487-13

FIGURE 13 CONVERSION OF THE DEEP-SET FROM CELL OUTPUT SYMBOLS TO CELL INPUT SYMBOLS



	A	B	C	D
1	1	–	–	–
2	2	1	2	1

(a) THE DEEP-SET WITH CELL INPUT SYMBOLS



INPUT:

	A	B	C	D	
1A	$1A^{\alpha}$	–	–	–	(1)
2A	$2A^{\alpha}$	$2B^{\alpha}$	$2C^{\alpha}$	$2D^{\alpha}$	(2)
2B	$1A^{\alpha}$	–	–	–	(1)
2C	$2A^{\gamma}$	$2B^{\gamma}$	$2C^{\gamma}$	$2D^{\beta}$	(3)
2D	$1A^{\alpha}$	–	–	–	(1)

(b) TRANSFORMATION FROM CELL INPUT SYMBOLS ON ONE DIAGONAL TO CELL OUTPUT SYMBOLS ON THE NEXT

INPUT:

	A	B	C	D
1	1^{α}	–	–	–
2	2^{α}	1^{α}	3^{α}	1^{α}
3	2^{γ}	1^{γ}	3^{γ}	1^{β}

(c) TRANSFORMATION, RELABELLED

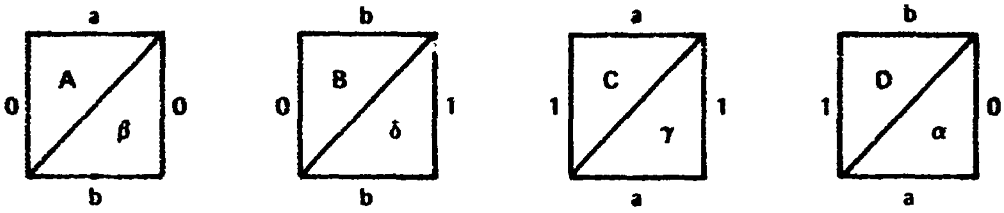
OUTPUT:

	α	β	γ	δ	
1	1	–	–	–	√
2	123	–	–	–	
3	–	1	123	–	
123	123	1	123	–	√

(d) THE OUTPUT SET

TA-8487-14

FIGURE 14 CONVERSION OF THE DEEP-SET FROM CELL INPUT SYMBOLS TO CELL OUTPUT SYMBOLS



(a) CELL OUTPUT AS A FUNCTION OF CELL INPUT

WEST CELL \ NORTH CELL:	α (a)	β (b)	γ (a)	δ (b)
$\alpha(0)$	β	δ	β	δ
$\beta(0)$	β	δ	β	δ
$\gamma(1)$	γ	α	γ	α
$\delta(1)$	γ	α	γ	α

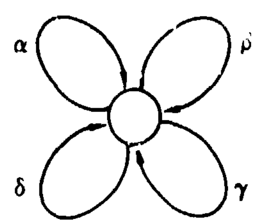
(b) CELL OUTPUT AS A FUNCTION OF NEIGHBORING CELL OUTPUTS

INPUT:	α	β	γ	δ
0	0^β	0^δ	1^β	1^δ
1	0^γ	0^α	1^γ	1^α

(c) THE TRANSDUCER, τ

OUTPUT:	α	β	γ	δ	
0	–	01	–	01	
1	01	–	01	–	
01	01	01	01	01	√

(d) THE OUTPUT TABLE



(e) THE k-LEVEL SET (FOR ALL k)

TA-8487-15

FIGURE 15 DERIVATION OF THE DEEP-SET FOR THE $\beta\delta\gamma\alpha$ EXAMPLE

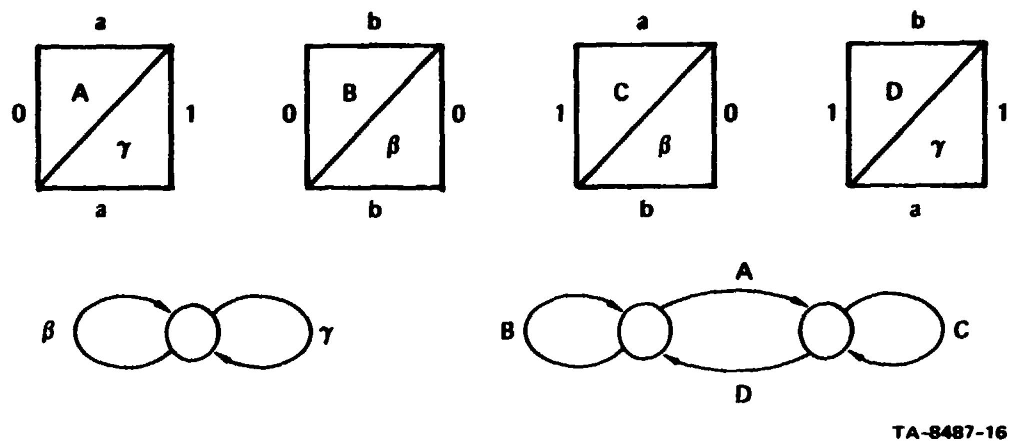


FIGURE 16 THE DEEP-SETS FOR THE $\gamma\beta\beta\gamma$ EXAMPLE

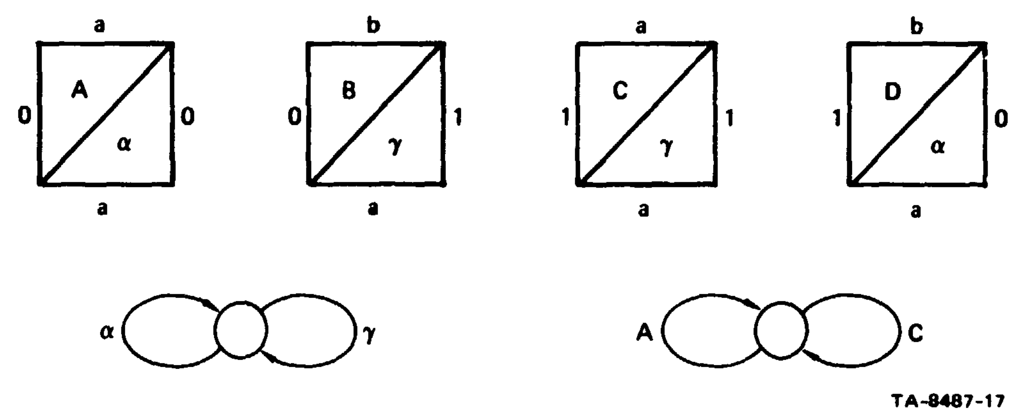
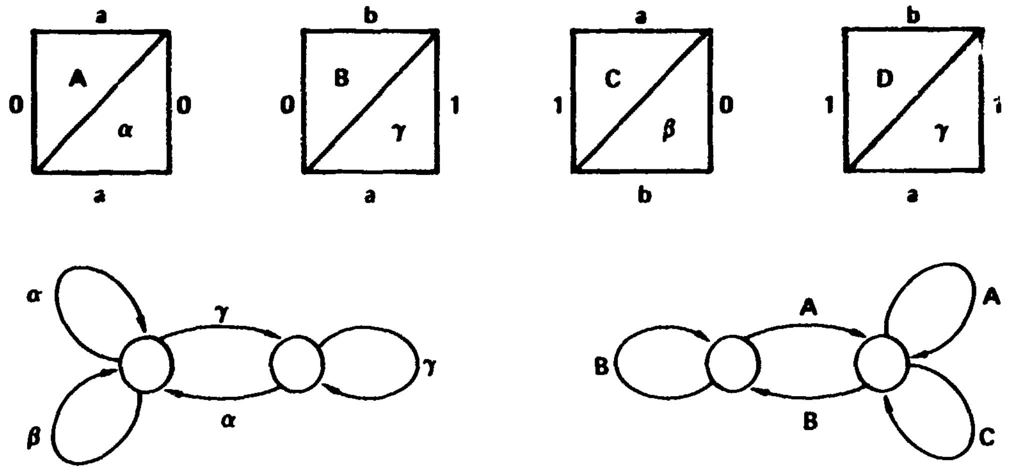
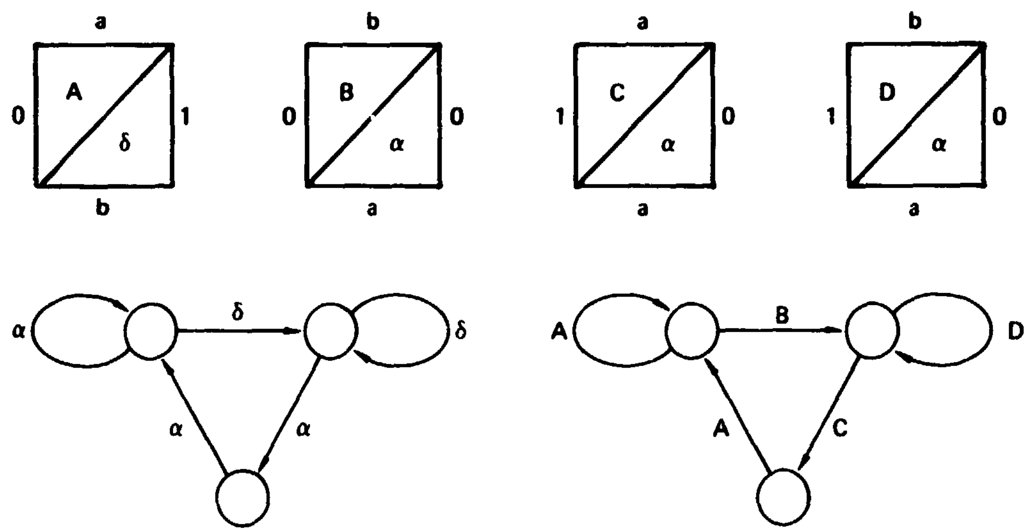


FIGURE 17 THE DEEP-SETS FOR THE $\alpha\gamma\gamma\alpha$ EXAMPLE



TA-8487-18

FIGURE 18 THE DEEP-SETS FOR THE $\alpha\gamma\beta\gamma$ EXAMPLE



TA-8487-19

FIGURE 19 THE DEEP-SETS FOR THE $\delta\alpha\alpha\alpha$ EXAMPLE

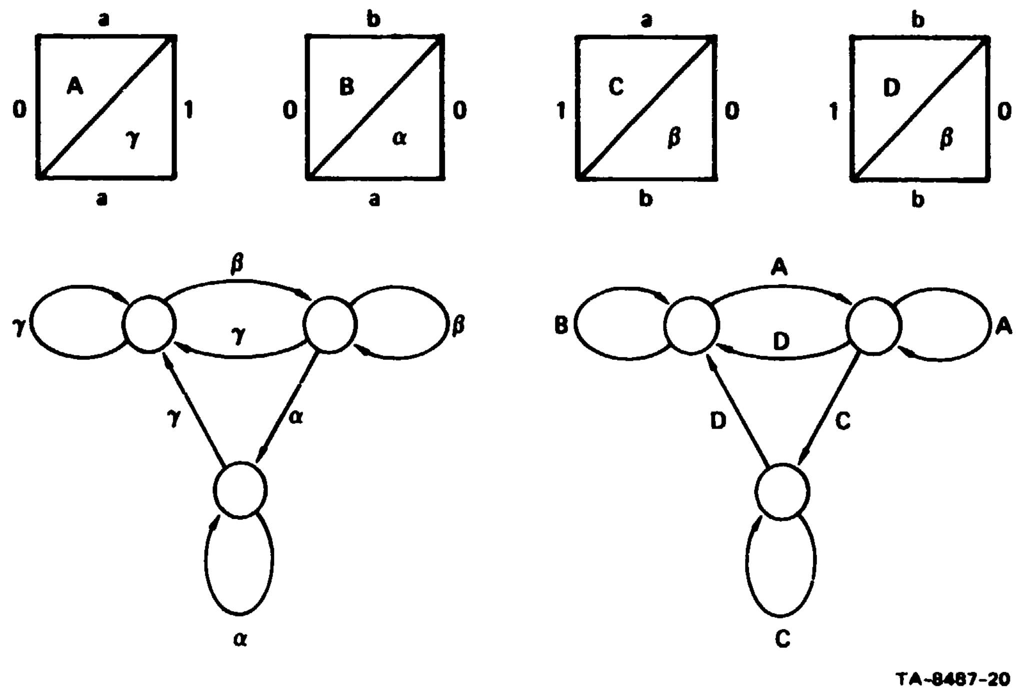


FIGURE 20 THE DEEP-SETS FOR THE $\gamma\alpha\beta\beta$ EXAMPLE

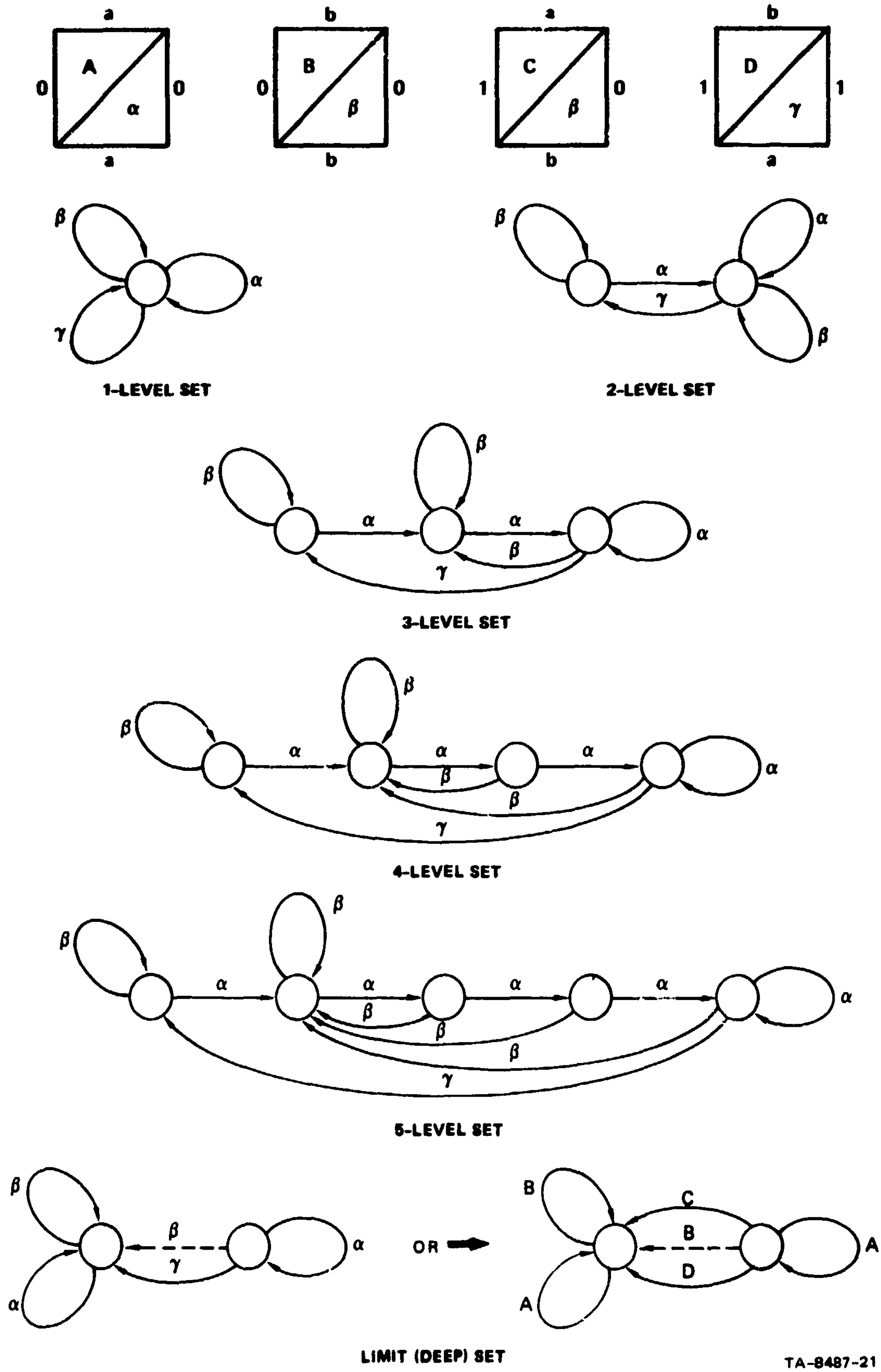


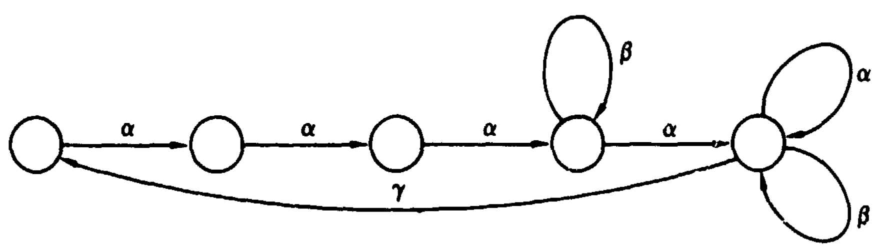
FIGURE 21 SEQUENCE OF DIAGONAL SETS FOR THE $\alpha\beta\beta\gamma$ EXAMPLE

	α	β	γ	δ
1	2	1	–	–
2	3	2	–	–
3	4	2	–	–
4	5	2	–	–
5	5	2	1	–

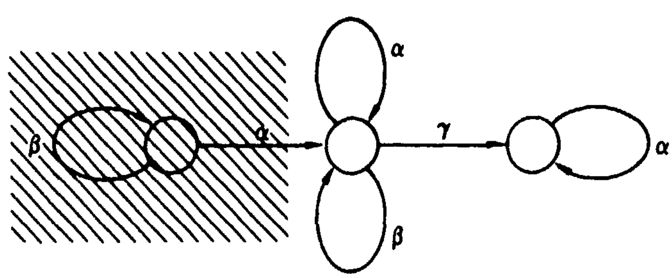
(a) THE 5-LEVEL SET

	α	β	γ	δ	
1	–	1	5	–	
2	1	2345	–	–	
3	2	–	–	–	
4	3	–	–	–	
5	45	–	–	–	√
45	345	–	–	–	√
345	2345	–	–	–	√
2345	12345	2345	–	–	√
12345	12345	12345	5	–	√

(b) DERIVATION OF THE REVERSED 5-LEVEL SET



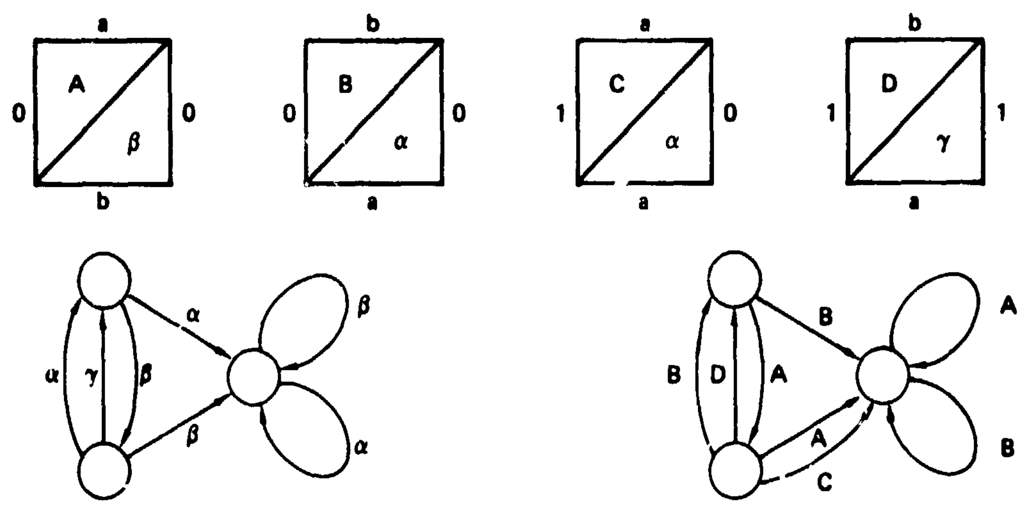
(c) THE REVERSED 5-LEVEL SET



(d) THE LIMIT OF THE SEQUENCE OF REVERSED SETS

TA-8487-22

FIGURE 22 REVERSED DIAGONAL SETS FOR THE $\alpha\beta\beta\gamma$ EXAMPLE

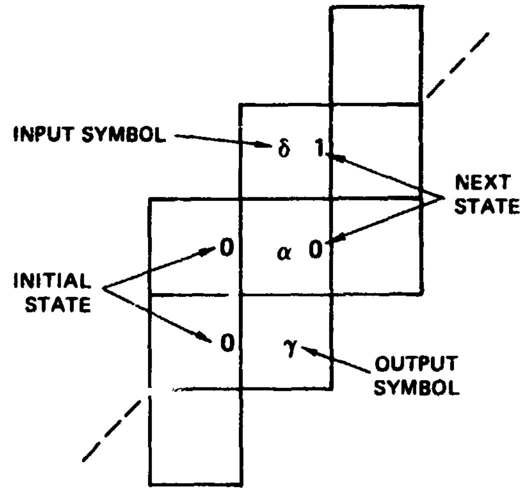


TA-8487-23

FIGURE 23 THE DEEP-SETS FOR A DIFFICULT EXAMPLE

	α	β	γ	δ
0	0^γ	0^α	1^γ	1^α
1	0^β	0^β	1^β	1^β

(a) THE TRANSDUCER, τ



(b) A TYPICAL STEP IN THE τ^2 TRANSFORMATION

	α	β	γ	δ
00	$0^\gamma 1^\gamma$	$0^\alpha 0^\gamma$	$1^\gamma 1^\gamma$	$1^\alpha 0^\gamma$
01	$0^\gamma 1^\beta$	$0^\alpha 0^\beta$	$1^\gamma 1^\beta$	$1^\alpha 0^\beta$
10	$0^\beta 0^\alpha$	$0^\beta 0^\alpha$	$1^\beta 0^\alpha$	$1^\beta 0^\alpha$
11	$0^\beta 0^\beta$	$0^\beta 0^\beta$	$1^\beta 0^\beta$	$1^\beta 0^\beta$

(c) THE TRANSDUCER, τ^2

	α	β	γ	δ
1	2^γ	1^γ	4^γ	3^γ
2	2^β	1^β	4^β	3^β
3	1^α	1^α	3^α	3^α
4	1^β	1^β	3^β	3^β

(d) τ^2, RENUMBERED

	α	β	γ	δ
1	2^β	1^γ	4^β	3^α
2	2^β	1^γ	4^β	3^α
3	1^γ	1^γ	3^α	3^α
4	1^γ	1^γ	3^α	3^α

(e) τ^2, WITH "SHIFTED" OUTPUTS

	α	β	γ	δ
1	1^β	1^γ	2^β	2^γ
2	1^γ	1^γ	2^α	2^α

(f) REDUCED FORM OF τ^2

TA-8487-24

FIGURE 24 STEPS IN THE DEVELOPMENT OF AN INTERDIAGONAL TRANSDUCER

	α	β	γ	δ	
10	$1^\beta 0^\alpha$	$1^\gamma 1^\gamma$	$2^\beta 0^\alpha$	$2^\alpha 0^\gamma$	√
11	$1^\beta 0^\beta$	$1^\gamma 1^\beta$	$2^\beta 0^\beta$	$2^\alpha 0^\beta$	√
20	$1^\gamma 1^\gamma$	$1^\gamma 1^\gamma$	$2^\alpha 0^\gamma$	$2^\alpha 0^\gamma$	√
21	$1^\gamma 1^\beta$	$1^\gamma 1^\beta$	$2^\alpha 0^\beta$	$2^\alpha 0^\beta$	

(a) THE TRANSDUCER, τ^3

	α	β	γ	δ
1	1^α	2^γ	3^α	3^γ
2	1^β	2^β	3^β	3^β
3	2^γ	2^γ	3^γ	3^γ

(b) $\tau^3 = \tau^6 = \tau^9 = \cdots$

	α	β	γ	δ
1	1^γ	2^β	3^γ	3^β
2	1^γ	2^α	3^γ	3^γ
3	2^β	2^β	3^β	3^β

(c) $\tau^4 = \tau^7 = \tau^{10} = \cdots$

	α	β	γ	δ
1	1^β	2^γ	3^β	3^α
2	1^β	2^γ	3^β	3^β
3	2^γ	2^γ	3^α	3^α

(d) $\tau^5 = \tau^8 = \tau^{11} = \cdots$

		FAULT TYPE:					
PRECEDING SEQUENCE:	STATE:	$\alpha-\beta$ $(\beta-\alpha)$	$\alpha-\gamma$ $(\gamma-\alpha)$	$\alpha-\delta$ $(\delta-\alpha)$	$\beta-\gamma$ $(\gamma-\beta)$	$\beta-\delta$ $(\delta-\beta)$	$\gamma-\delta$ $(\delta-\gamma)$
$\alpha/\beta-\alpha$	1	ANY	α/γ	ANY	ANY	ANY	ANY
OR $\gamma/\delta-\left.\begin{matrix}\beta\\\alpha\end{matrix}\right\}$	2	ANY	α/γ	α/γ	ANY	ANY	NONE
γ/δ	3	NONE	ANY	ANY	ANY	ANY	NONE

(e) CONTINUATIONS OF DIAGONAL SEQUENCES WHICH WILL PROPAGATE ERRORS

TA-8487-25

FIGURE 25 ERROR PROPAGATION ANALYSIS FOR THE $\gamma\alpha\beta\beta$ EXAMPLE

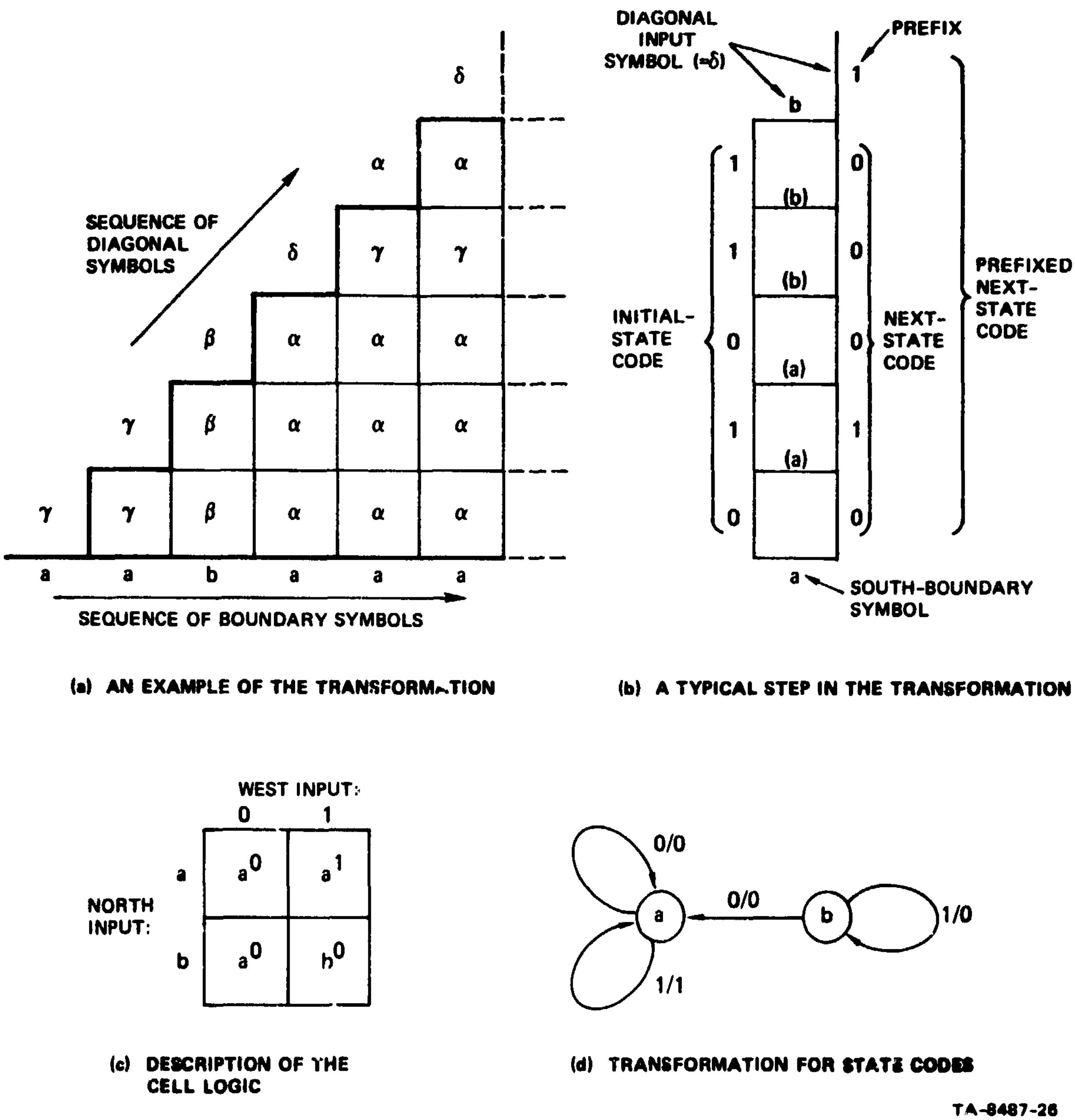
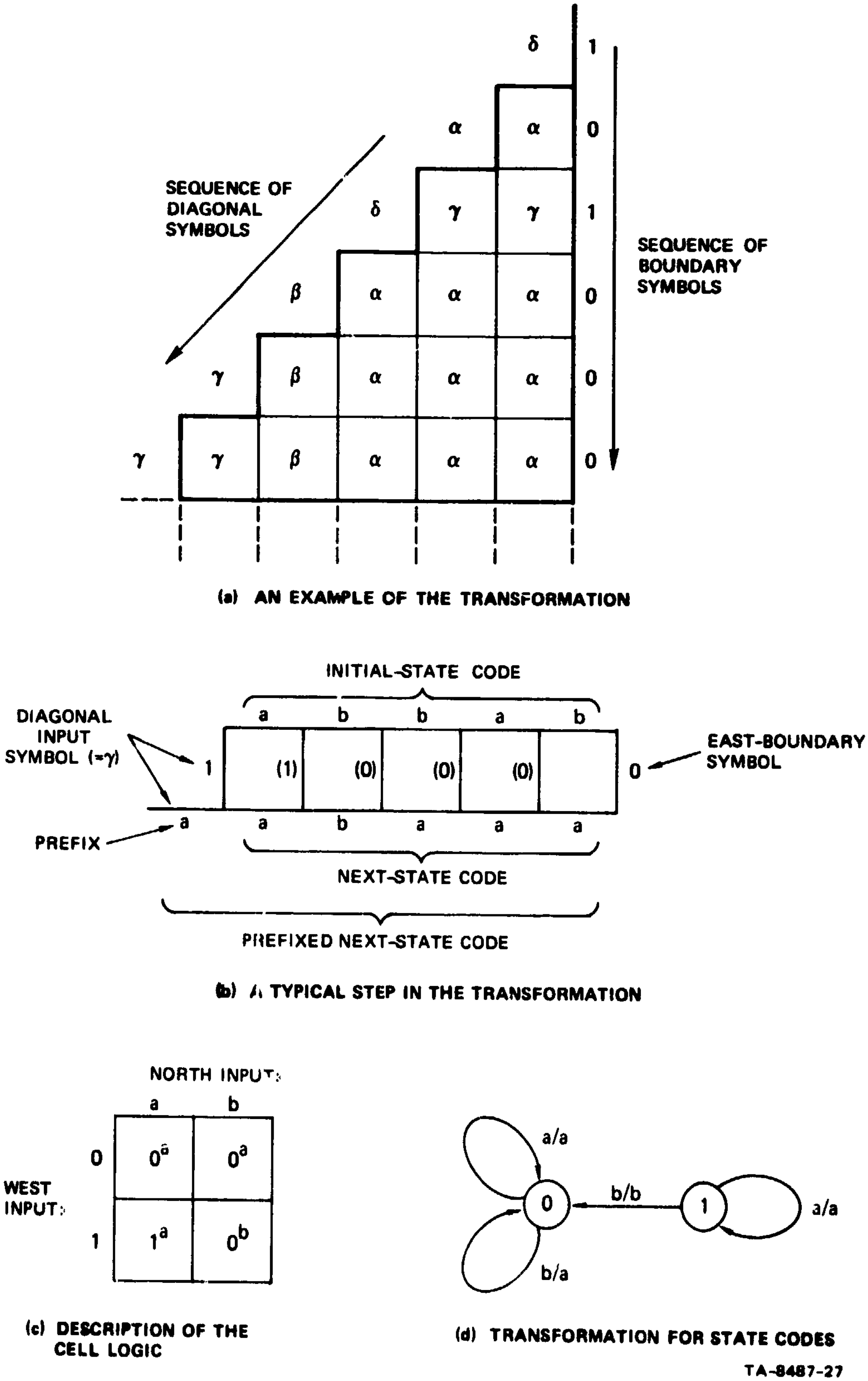


FIGURE 26 STEPS IN THE DERIVATION OF A DIAGONAL-TO-BOUNDARY TRANSDUCER FOR THE $\alpha\alpha\gamma\beta$ EXAMPLE



(a) AN EXAMPLE OF THE TRANSFORMATION

(b) A TYPICAL STEP IN THE TRANSFORMATION

(c) DESCRIPTION OF THE CELL LOGIC

(d) TRANSFORMATION FOR STATE CODES

FIGURE 27 STEPS IN THE DERIVATION OF ANOTHER DIAGONAL-TO-BOUNDARY TRANSDUCER FOR THE $\alpha\alpha\gamma\beta$ EXAMPLE

	α	β	γ	δ
1	1	–	–	–
2	2	1	2	–

(a) THE DEEP-SET

	α	β	γ	δ
→ 1	3^a	3^b	2^a	2^b
2	3^a	3^b	2^a	3^b
3	3^a	3^a	2^a	3^a

(b) THE DIAGONAL-TO-SOUTH BOUNDARY TRANSDUCER

	α	β	γ	δ
1	1	2	–	–
2	2	–	2	–

(c) THE DEEP-SET, REVERSED

	α	β	γ	δ
→ 1	1^0	2^0	1^1	2^1
2	1^0	2^0	2^0	2^0

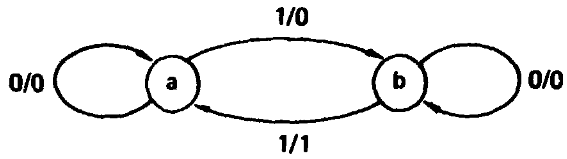
(d) THE DIAGONAL-TO-EAST-BOUNDARY TRANSDUCER

TA-8487-28

FIGURE 28 TABLES FOR THE ANALYSIS OF ERROR PROPAGATION FOR THE $\alpha\alpha\gamma\beta$ EXAMPLE

	0	1
a	a^0	b^0
b	b^0	b^1

(a) CELL LOGIC



(b) TRANSFORMATION FOR STATE CODE

	α (a0)	β (b0)	γ (a1)	δ (b1)
k EVEN	k/2	k/2	(k+2)/2	(k+2)/2
k ODD	(k−1)/2	(k+1)/2	(k+1)/2	(k+3)/2

(c) COMPUTATION OF THE NUMBER OF 1s IN THE NEXT-STATE CODE

	α	β	γ	δ
→ 0	0^a	0^b	1^a	1^b
1	0^b	1^a	1^b	2^a
2	1^a	1^b	2^a	2^b

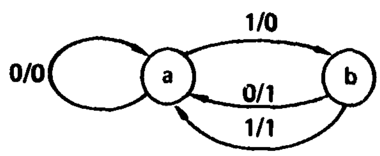
(d) THE DIAGONAL-TO-SOUTH-BOUNDARY TRANSDUCER

TA-8487-29

FIGURE 29 TABLE FOR ERROR PROPAGATION ANALYSIS OF THE $\alpha\beta\beta\gamma$ (HALF ADDER) EXAMPLE

	0	1
a	a^0	b^0
b	a^1	a^1

(a) CELL LOGIC



(b) TRANSFORMATION TO NEXT-STATE CODE

	α (a0)	β (b0)	γ (a1)	δ (b1)
00 · · ·	0000···	0100···	1000···	1100···
01 ···	0001···	0101···	1001 ···	1101···
10 ···	0010···	0100 ···	1010···	1100···
11 ···	0010 ···	0101 ···	1010 ···	1101 ···

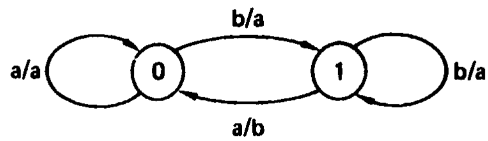
(c) TABLE SHOWING TRANSFORMATION FROM CODE FOR INITIAL STATE TO CODE FOR PREFIXED NEXT-STATE

TA-8487-30

FIGURE 30 ANALYSIS OF ERROR PROPAGATION TO THE SOUTH ARRAY BOUNDARY FOR THE $\alpha\gamma\beta\gamma$ EXAMPLE

	a	b
0	0^a	1^a
1	0^b	1^a

(a) CELL LOGIC



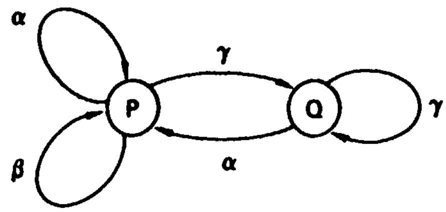
(b) TRANSFORMATION TO NEXT-STATE CODE

	α (a0)	β (b0)	γ (a1)	δ (b1)
aa···	aaa···	baa···	aba···	bba···
ab···	aaa···	baa···	aba···	bba···
ba···	aab···	bab···	aab···	bab···
bb···	aaa···	baa···	aaa···	baa···

(c) TABLE SHOWING TRANSFORMATION FROM CODE FOR INITIAL STATE TO CODE FOR PREFIXED NEXT-STATE

TA-8487-31

FIGURE 31 ANALYSIS OF ERROR PROPAGATION TO THE EAST ARRAY BOUNDARY FOR THE $\alpha\gamma\beta\gamma$ EXAMPLE



(a) THE DEEP-SET

	α	β	γ	δ
→ P	P	P	Q	–
Q	P	–	Q	–

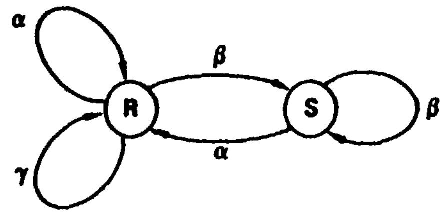
(b) THE DEEP-SET

	α	β	γ	δ	
P00	$P00^{00}$	$P01^{00}$	$Q10^{00}$	-11^{00}	√
P01	$P00^{01}$	$P01^{01}$	$Q10^{01}$	-11^{01}	√
P10	$P00^{10}$	$P01^{00}$	$Q10^{10}$	-11^{00}	
P11	$P00^{10}$	$P01^{01}$	$Q10^{10}$	-11^{01}	
Q00	$P00^{00}$	-01^{00}	$Q10^{00}$	-11^{00}	
Q01	$P00^{01}$	-01^{01}	$Q10^{01}$	-11^{01}	
Q10	$P00^{10}$	-01^{00}	$Q10^{10}$	-11^{00}	√
Q11	$P00^{10}$	-01^{01}	$Q10^{10}$	-11^{01}	

(c) TABLE SHOWING PROPAGATION FROM DEEP CELLS

TA-8487-32

FIGURE 32 DERIVATION OF A COMPOSITE TABLE SHOWING ERROR PROPAGATION TO THE SOUTH BOUNDARY FROM DEEP CELLS FOR THE $\alpha\gamma\beta\gamma$ EXAMPLE



(a) THE REVERSED DEEP-SET
(reduced to deterministic form)

	α	β	γ	δ
→ R	R	S	R	–
S	R	S	–	–

(b) THE REVERSED DEEP-SET

	α	β	γ	γ	
Raa	Raa^a	Sba^a	Rab^a	$-bb^a$	√
Rab	Raa^a	Sba^a	Rab^a	$-bb^a$	√
Rba	Raa^b	Sba^b	Raa^b	$-ba^b$	
Rbb	Raa^a	Sba^a	Raa^a	$-ba^a$	
Saa	Raa^a	Sba^a	$-ab^a$	$-bb^a$	
Sab	Raa^a	Sba^a	$-ab^a$	$-bb^a$	
Sba	Raa^b	Sba^b	$-aa^b$	$-ba^b$	√
Sbb	Raa^a	Sba^a	$-aa^a$	$-ba^a$	

(c) TABLE SHOWING PROPAGATION FROM DEEP CELLS

TA-8487-33

FIGURE 33 DERIVATION OF A COMPOSITE TABLE SHOWING ERROR PROPAGATION TO THE EAST BOUNDARY FROM DEEP CELLS FOR THE $\alpha\gamma\beta\gamma$ EXAMPLE